Efficient Algorithms for Convex Optimization based Control

Yue Yu

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

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Behçet Açıkmeşe, Chair

Maryam Fazel

Mehran Mesbahi

Program Authorized to Offer Degree:
Aeronautics & Astronautics

University of Washington

Abstract

Efficient Algorithms for Convex Optimization based Control

Yue Yu

Chair of the Supervisory Committee:
Professor Behçet Açıkmeşe
William E. Boeing Department of Aeronautics & Astronautics

This dissertation studies efficient optimization algorithms designed to solve control problems. It is composed of the following three parts.

- Part I: This part focuses on Markovian network equilibrium, a novel class of stochastic dynamic network equilibrium problems. After introducing the problem formulation, we discuss efficient dynamic-programming-based algorithms designed for these optimization problems.

- Part II: This part focuses on first order convex optimization methods for distributed optimization and trajectory optimization. The key idea is combining proportional-integral feedback with projected gradient or mirror descent method.

- Part III: This part focuses on Willems' fundamental lemma, a key result in system identification and data-driven control. We generalized previous results to handle uncontrollable systems and systems with special structures.

Table of Contents

List of Figures

Figure Number Page

Acknowledgments

I wish to thank my advisor, Professor Behçet Açıkmeşe. There are not enough words in my vocabulary to describe how much I value our time together. Many lessons I learned from him has changed my life completely and hard coded in my brain. For example, I always take detailed notes every time I have a research idea–however immature they look at the time–just the way he does. When I talk to my colleagues, I am never shy of showing my excitement for new ideas, because I know that the best way to inspire people is to show passion –a lesson that Behçet demonstrated to me countless times. More importantly, I will never forget how much help Behçet provides for his students, in both research and personal life. There has been some very uncertain and disturbing time during my PhD years, but I hardly worry about anything other than research. The reason is simply because I know Behçet will be there to help me, and he is willing to move mountains for his students. Because of these priceless qualities, Behçet attracted many brilliant young minds to the Autonomous Controls Laboratory, which also benefits my research tremendously over the years. In one sentence, I will die a happy man if my future advisees think of me the same way I think of Behçet.

I also want to thank Professor Mehran Mesbahi. Among all the people I have ever known, Mehran is the best conversationalist–a conversation with him is like a session with a professional therapist. He also has the largest knowledge volume I have ever seen. Whenever I get stuck with a research puzzle, he can always get me what I need: sometimes they are names or keywords I never heard of, other times it is a book in his personal library. Perhaps what I appreciate the most in Mehran is that he is very fashionable–in his wardrobe too, but more impressively, in his research. He always manages to stay on top of the recent trends in research, rather than growing stale like one would expect from a typical knowledgeable senior professor. In one sentence, I think of Mehran as one of the best role model in academia world.

Dan Calderone, Sarah H.Q. Li, and Professor Lillian J. Ratliff, it has been a extremely precious experience to work with you all. Dan, thank you for teaching me all about Wardrop equilibrium. I cannot remember how many hours of crash-course lectures you gave me; they are among the most valuable memories during my PhD. Sarah, your curiosity and passion about research rekindled mine so many times. And I cannot be more proud about what you acomplished. Lily, thank you for serving on my committee since my qualifying exams, and for the conversations we had on academic career planning during the bus rides we shared.

Purnanand Elango, working with you is definitely one of the highlights in my PhD. You always say that you benefit from our collaboration more than I do, since you always learn some new knowledge from me. But one big secret in our profession is that the best catalyst for good research is not old knowledge, but rather the time and attention of a curious, intelligent, and patient mind. For that reason, our collaboration is just as valuable to me as it is to you. Looking forward to continue our journey together!

Danylo Malyuta, it has been a really rewarding experience to work with you. The clarity of your thinking, the expertise and enthusiasm you have on aerospace engineering, your magical speed of absorbing new knowledge, and all the other amazing qualities of yours have impressed me as well as everyone around you. I am sure the future holds great things for you. Go change the world!

Henk van Waard and Shahriar Talebi, I am so glad that we found each other, and started our current work. Henk, your PhD work is a real eye-opener, it completely changed the way I think of data-driven control. Like I said at your graduation ceremony, I am excited to continue our work together. Shahriar, I have always valued you as much a research partner as a close confidant. Solving math puzzles with you is one of the most fun part of my daily life. Looking forward to start the next one together!

Mahmoud El Chamie, thank you for mentoring me when I first started; my first year would have been a lot more difficult without you. Michael Szmuk and Skye Mceowen, thank you for building such an amazing lab: it is a Disneyland for someone who wants to test a control algorithm. Daniel Dueri, Utku Eren, Nazli Demir, Dylan Janak, Sean Rice, Xiangru Xu, Mathias Hudoba de Badyn, Airlie Chapman, Siavash Alemzadeh, Taylor Reynolds, Dillon Foight, Bijan Barzgaran, and everyone else that I ever interacted with at the University of Washington, thank you all for the good memories we shared.

I also want to give a special thank you to Professor Maryam Fazel, Professor Rekha R. Thomas, Professor James V. Burke, Professor Aleksandr Aravkin, Professor Yin Tat Lee, and Professor Dmitriy Drusvyatskiy, for the fantastic lectures on optimization they give and the amazing seminars they help organize.

Finally, I would like to thank my mother, Shulan Yu for raising me and constantly caring for me from thousands of miles away. I want to thank my girlfriend and best friend, Wenying Zhang. Everyday I cannot believe how lucky I am to have you in my life.

Dedication

to Shulan Yu and Wenying Zhang

1 Welcome, my dear reader

In view of the extremely small number of people who still read Ph.D dissertation, I am thrilled that you are reading this very sentence. So what is in this dissertation? It contains three separate parts, each belongs to one sub-domain in optimization-based control theory: network equilibrium, network optimization, data-driven control. Each part starts with a motivating example, followed by some key results, then finishes with literature review on related work. In this way, you can not only learn the original results in this dissertation without heavy reading assignments, but also still be able to track down related references if needed. Finally, the Appendix includes relevant details used in this dissertation, which offers some crash course materials on basic convex optimization and control theory.

Here are the outlines of each individual part.

1.1 Outline of Part I

This part focuses on network equilibrium problems, an intersection of network optimization and game theory. These problems arise in a variety of applications, such as transportation, economics, as well as communication; see [150, 17, 137] for some early references. Our main theoretical tool is monotropic optimization theory [150, 17].

We will introduce a novel class of network equilibrium problem, which combines Markov decision process (MDP), a powerful stochastic decision making model, and Wardrop equilibrium, a key concept in multi-player network games. First, we will revisit the linear programming formulation of MDP [141] in §2.1.1, and bring it into the network optimization framework [150]. Second, we will introduce the network equilibrium extension to MDP, termed Markovian network equilibrium in §2.1.2, along with its variations in variable demand and multi-commodity scenarios, discussed in §2.2. Each section not only shows how these equilibrium problems can be modeled as convex optimization, but also provides dynamic-programming-based iterative algorithms, which outperform state-of-the-art optimization software significantly, as we demonstrate in §2.3. Finally, §2.4 briefly reviews some classical results in network equilibrium, some relevant literature in game theory, and comments on future work.

The main references for this part are [30, 29, 31, 193, 106].

For background materials related to this section, see the section on the KKT conditions in Appendix A.2.2, projected subgradient method in Appendix B.2, and conditional gradient method in Appendix B.5.

1.2 Outline of Part II

This part focuses on the design and analysis of first order optimization methods for distributed optimization, one of the most well-studied class of optimization problem during the past decade [18, 25], and convex trajectory optimization, the fundamental optimization problem in model predictive control [120, 119, 142, 91, 55]. Our algorithm design and analysis is based on discretization of ordinary differential equations, which recently has been a popular approach; see [161, 94, 180, 45].

We will start with distributed optimization in §3.1. First, we revisit some classical results for distributed consensus, a simplified version of distributed optimization, in §3.1.1, where we use the RC circuits analogy to interpret distributed consensus protocol [122]. Based on this interpretation, we introduce a novel distributed optimization algorithm inspired by RLC circuits dynamics in §3.1.2. We also discuss its extensions to noisy gradients and composite objective settings. Next, we use the similar idea to design a novel first order primal-dual method for convex trajectory optimization problem, named PI projected gradient method, in §3.2. Considering their rich literature, we devote §3.3 to the related work on distributed optimization algorithms, and §3.2 to the related work on convex trajectory optimization algorithms.

The main references for this part are [191, 192, 190, 194].

For background materials, see optimality conditions in Appendix A, mirror descent method in Appendix B.4, projected subgradient method in Appendix B.2, projected gradient method in Appendix B.1.

1.3 Outline of Part III

This part focuses on a key result in system identification and data-driven control, the so-called *fundamental lemma* or *Willems' fundamental lemma*. This lemma was first proved in [182], and recently extended to multiple data trajectories setting in [166].

Our contribution is to prove more general results than those in [166] by relaxing their assumptions. We prove our main result, a generalization of Willems' lemma, in §4.1. Our results show that online data is particularly useful in data-driven predictive control of uncontrollable systems, and structures of homogeneous multi-agent system, which are demonstrated via numerical examples in §4.2.

The main references for this part are [182, 166].

For background materials, see Appendix D.

1.4 Building an optimization course

Do you want to teach a graduate-student-level course on convex optimization someday? If so, this dissertation will help you with your lecture notes on many topics. This section will give some guidelines on how to use different sections of the dissertation to build a toy syllabus. The reader is encouraged to adjust the organization based on his/her specific interests[1].

In particular, materials in Part I and Part II can seamlessly fit into the framework of convex optimization. Further, compared with some existing materials[2], Appendix A provides a uniform perspective on optimality conditions; Appendix B provides a uniform "4-step formul" for writing convergence proof of many classical first order convex optimization algorithms. A potential outline for a "toy syllabus" is given as follows.

[1]During his PhD years, the author benefited tremendously from the optimization lectures offered by Prof. Maryam Fazel, Prof. Rekha R. Thomas, Prof. James V. Burke, Prof. Yin Tat Lee, Prof. Aleksandr Aravkin, and Prof. Dmitriy Drusvyatskiy. This section is dedicated to their excellent work in teaching.

[2]The author's favorite books/tutorial papers include [152, 23, 26].

1. Model

    (a) This part starts with basics of convex sets, functions and optimization, followed by KKT conditions and Lagrangian duality theory. See Appendix A and [23] for necessary materials.

    (b) Next, §2.4.1.1 and §2.1.1 provide examples of KKT conditions of linear programming from network equilibrium problems, and the procedure of solving these conditions directly using dynamic programming; see §2.1.1.

    (c) Finally, §2.4.1.2 and §2.1.2 show how to solve special network equilibrium problems using projected subgradient method in Appendix B.2 and conditional gradient method in Appendix B.5, which motivates iterative first order convex optimization methods in the next part.

2. Methods

    (a) This part starts with the classical convergence results of projected gradient descent in Appendix B.1, projected subgradient method in Appendix B.2, proximal gradient method in B.3, and conditional gradient method in Appendix B.5.

    (b) Next, the teacher can discuss extensions to mirror descent in Appendix B.4. Appendix B.6 provides more recent results on the accelerated mirror descent method.

    (c) §3 provides examples of recent results on first order methods for distributed optimization in §3.1 and trajectory optimization in §3.2. §4 provides extended reading materials on data-driven trajectory optimization.

1.5 Notations

Let $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of real numbers, nonnegative real numbers and, respectively, positive real numbers; $\mathbb{R}^n$ denotes the n-dimensional real numbers; $\mathbb{N}$ denotes the set of positive integers. Let $\cdot^\top$ denote the matrix (and vector) transpose, $\langle x, y\rangle = x^\top y$ and

$\|x\|_2 = \sqrt{\langle x, x\rangle}$ the inner product and, respectively, ℓ_2 norm. Let $\mathbf{diag}(x) \in \mathbb{R}^{n\times n}$ denote the diagonal matrix with diagonal elements $x \in \mathbb{R}^n$, I_n the $n \times n$ identity matrix, $\mathbf{1}_n \in \mathbb{R}^n$ the vector of all 1's, $\otimes$ the Kronecker product, and $\mathbb{E}[\cdot]$ the expectation.

2 Markov decision process and network equilibrium

Imagine the following scenario. The city of Seattle is considering building a new light rail transit (LRT) route in its existing transportation network, illustrated in Fig. 2.1. Right now the mayor is considering two candidate routes, both marked in Fig. 2.1. Suddenly she turned to you and asks the question “Is there a mathematical model that tells which plan is better in terms of alleviating the traffic congestion in the city?"

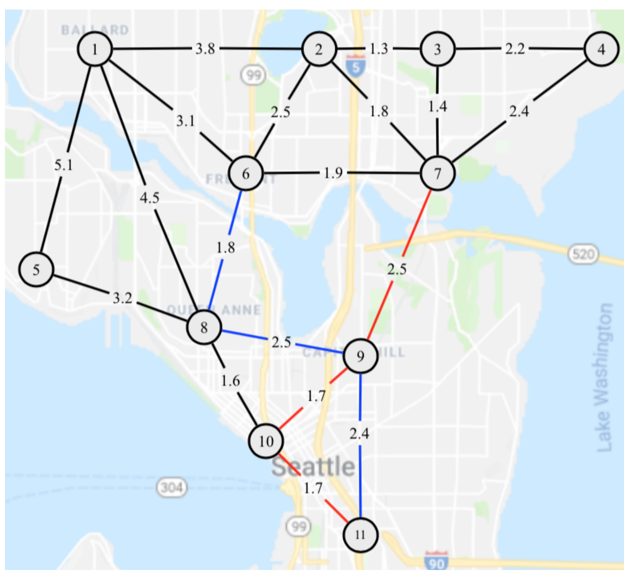


Figure 2.1: Seattle transportation network and candidate LRT route 7-9-10-11(red) and 6-8-9-11(blue). Each link is labeled by the corresponding driving distance.

The answer to the above question is a research topic that has been studied in the context of operations research, convex optimization, and game theory, under the name of Wardrop equilibrium [177], traffic assignment problem [137, 61, 62], network equilibrium/optimization problem [150, 17], and routing games [155]. Here we adopt the name *network equilibrium*. The key idea of these existing work is first model each user's decison-making in the transportation network as a shortest path problem, then approximate the behavior of the entire user population using the so-called *Wardrop equilibrium principle*.

So what is new in this chapter?

We will replace the shortest path problem, a static deterministic decision-making model, in the existing literature with a dynamic and stochastic one, known as the *Markov decision process* (MDP) [141, 16]. MDP not only includes the shortest path problem as a special case, but also naturally handles uncertainties arises from, *e.g.*, model parameter estimation. Our goal is to show how the network equilibrium theory looks like with this key replacement. The main mathematical tool used in this chapter is the notion of Lagrangin and KKT conditions introduced in Appendix A. In fact, this entire chapter can be summarized as: applications of KKT conditions in a special class of convex optimization problems where the duality theory is as sharp as the one for linear programming[1].

2.1 Markovian network equilibrium

This section introduces the key results of this chapter, which solve the aforementioned transportation network design problem. In particular, §2.1.1 introduces basic concepts in MDP, which models the decision-making of an individual user in a given network. §2.1.2 extends MDP to a general network equilibrium model, which captures the behavior of a population of users. Together, they provide a mathematical model that captures the user behavior in a transportation network, and predicts how the user patterns shift when certain part of the network is modified, *e.g.*, by a new LRT route. After discussing two further variations of this basic model in §2.2, we illustrate how our results solve the transportation network design problem numerically in §2.3.

[1]According to Tyrrell Rockafellar [152, p. vi] and Dimitri Bertsekas [17, p. 408].

2.1.1 Markov decision process

Markov decision process (MDP) is a discrete time stochastic control process, where a decision maker tries to minimizes the expected accumulated costs by choosing a sequence of actions in a stochastic environment. A finite horizon MDP is defined by the following components

- time index: $[T] = \{1, 2, \ldots, T\}$
- states: $[S] = \{1, 2, \ldots, S\}$
- actions: $[A] = \{1, 2, \ldots, A\}$
- transition probability: $P \in [0, 1]^{S\times A\times S}$, such that $\sum_{s'} P_{sas'} = 1$ for all $s \in [S], a \in [A]$
- immediate costs: $c \in \mathbb{R}^{T\times S\times A}$

With these notions, a T-horizon MDP is defined as follows.

Problem 2.1. *Consider the following stochastic decision process. At each time $t \in [T]$, the decision maker can choose to take action a in state s at time t at the cost of c_{tsa}, then reach state $s' \in [S]$ with probability $P_{sas'}$ at time $t + 1$, then repeat such process till $t = T$, when the decision process ends after choosing the last action. Starting from an arbitrary given state $s_0 \in [S]$ at time $t = 1$, find an optimal sequence of actions that minimizes the expected accumulated costs over time window $[T]$.*

Different solution methods have been developed for Problem 2.1, and its infinite-horizon extension. Among these methods, linear programming approach has attracted increasing attention due to its elegance in theory and compatibility with constraints. In particular,

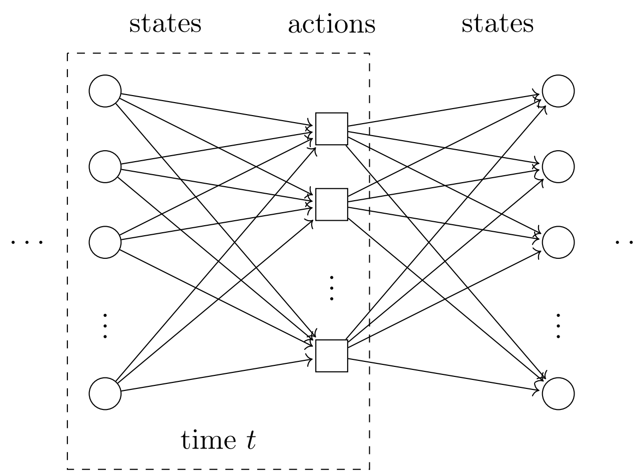


Figure 2.2: A illustration of MDP

Problem 2.1 can be equivalently modeled as the two following linear programming

$$\begin{aligned}
\underset{y}{\text{minimize}} \quad & \sum_{t,s,a} c_{tsa} y_{tsa} \\
\text{subject to} \quad & \sum_a y_{1sa} = p_{1s}, \\
& \sum_a y_{t+1,sa} = p_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a}, \;\; t \in [T-1], \\
& y_{tsa} \geqslant 0, \;\; \forall t \in [T], s \in [S], a \in [A]
\end{aligned} \tag{2.1}$$

$$\begin{aligned}
\underset{v}{\text{maximize}} \quad & \sum_{t,s} p_{ts} v_{ts} \\
\text{subject to} \quad & v_{Ts} \leqslant c_{Tsa} \\
& v_{ts} \leqslant c_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}, \\
& \forall t \in [T-1], s \in [S], a \in [A].
\end{aligned} \tag{2.2}$$

where $y \in \mathbb{R}^{T\times S\times A}$ is called the *state-action-frequency*; $v \in \mathbb{R}^{T\times S}$ is called the value, or *cost-to-go*.

We group our assumptions on linear programming (2.1) and (2.2) as follows.

Assumption 2.1. *We assume* $p \in \mathbb{R}_+^{T\times S}$, $P \in \mathbb{R}_+^{S\times A\times S}$ *and* $\sum_{s'} P_{sas'} = 1$ *for all* $s \in [S]$,

$a \in [A]$.

How does the solution to (2.1) and (2.2) help solve Problem 2.1? The answer is illustrated in the following theorem; we will use the following short-hand notation: given $b_1, \ldots, b_N \in \mathbb{R}$,

$$(b^\star, i^\star) = \min_{i \in [N]} \; b_i \iff \quad b^\star = \min_{i \in [N]} \; b_i, \;\; i^\star \in \operatorname*{argmin}_{i \in [N]} \; b_i. \tag{2.3}$$

Theorem 2.1. *Suppose Assumption 2.1 holds, y solves* (2.1)*, and v solves* (2.2)*. If $y_{tsa} > 0$, then*

$$\begin{aligned} (v_{Ts}, a) &= \min_{a' \in [A]} \; c_{Tsa'}, \\ (v_{ts}, a) &= \min_{a' \in [A]} \; c_{tsa'} + \sum_{s'} P_{sa's'} v_{t+1,s'}, \end{aligned} \tag{2.4}$$

for all $t \in [T-1]$, where we use the convention in (2.3)*.*

Proof. It is straightforward to verify that the solution set of (2.1) and (2.2) are non-empty. Hence a solution pair to (2.1) and (2.2) necessarily satisfy the KKT conditions [152, Thm. 28.3.1]. Let v_{ts} be the dual variable corresponding to the equality constraints containing p_{ts}, let $\mu_{tsa}, \theta_{ts}, \lambda_{ts} \geqslant 0$ be the dual variables corresponding to constraint $y_{tsa} \geqslant 0$. Then the Lagrangian of(2.1) and (2.2) is given by

$$L(y, v, \mu) = \sum_{t,s,a} (c_{tsa} - \mu_{tsa}) y_{tsa} + \sum_{t,s,a} v_{ts}(p_{ts} - y_{tsa}) + \sum_{t, t<T} \sum_{s', a, s} v_{t+1,s} P_{s'as} y_{ts'a}.$$

The KKT conditions [152, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L / \partial y_{tsa}$ equal to zero)

$$\begin{aligned} v_{Ts} =& c_{Tsa} - \mu_{Tsa}, \\ v_{ts} =& c_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa}, \;\; t \in [T-1], \end{aligned} \tag{2.5}$$

for all $s \in [S], a \in [A]$, and the complementarity conditions

$$y_{tsa}\mu_{tsa} = 0, \;\; y_{tsa} \geqslant 0, \;\; \mu_{tsa} \geqslant 0, \;\; \forall t \in [T], s \in [S], a \in [A]. \tag{2.6}$$

Combining (2.5) and (2.6) gives the desired results. Note that same results can be derived from the dual problem (2.2). □

Theorem 2.1 shows how the solution to (2.1) and (2.2) can be used to solve Problem 2.1. In particular, action a is optimal in state s at time t if

- $y_{tsa} > 0$, where y solves (2.1),
- $v_t s = c_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}$, where v solves (2.2).

In other words, solutions to (2.1) or (2.2) provides a *optimal policy* that maps time-state pairs to the optimal action that minimizes expected cost-to-go. Once a solution to (2.1) or (2.2) is obtained, one can solve Problem 2.1 by repeatedly query the optimal policy during the decision process. In other words, in order to solve Problem 2.1, it suffices to solve either (2.1) or (2.2).

Therefore, our remaining question is: how to solve linear programming (2.1) and (2.2)? Many popular algorithms have been developed for general linear programming problems. However, due to the special structure of their KKT conditions (see the proof of Theorem 2.1), (2.1) and (2.2) can be solved in closed-form using *dynamic programming*. In particular, using Lemma A.4, one can verify that the output v of Algorithm 1 is an optimal solution to (2.2); in addition, if the input π in Algorithm 2 is the output of algorithm 1, then its output y of Algorithm 2 and obtain output y is an optimal solution to (2.1).

The readers may be wondering: if we already obtain the optimal policy after executing Algorithm 1, why bothering with Algorithm 2 at all? Well, in fact, most literature on MDP pays much very little attention to Algorithm 2, for this exact reason. However, when we extend Problem 2.1 to a general equilibrium problem, Algorithm 2 becomes a subroutine every bit as important as Algorithm 1, as we will show next.

Algorithm 1 Backward induction

Input: P, c, T.

Output: v, π.

Let $(v_{Ts}, \pi_{Ts}) = \min_a c_{Tsa}, \ \forall s \in [S]$.

for $t = T-1, T-2, \ldots, 1$ **do**

$\quad (v_{ts}, \pi_{ts}) = \min_{a \in [A]} \big(c_{tsa} + \sum_j P_{saj} v_{t+1,j} \big), \ \forall s \in [S]$

end for

Algorithm 2 Forward induction

Input: π, p, P, T.

Output: y.

Initialize $y = 0$, let $y_{1s\pi_{1s}} \leftarrow p_{1s}$ for all $s \in [S]$.

for $t = 1, 2, \ldots, T-1$ **do**

$\quad y_{t+1,s\pi_{t+1,s}} \leftarrow p_{t+1,s} + \sum_j P_{j\pi_{tj}s} y_{tj\pi_{tj}}, \ \forall s \in [S]$

end for

2.1.2 Wardrop equilibrium

Now let us turn our attention back to the transportation network problem introduced at the beginning of this chapter. The MDP model in §2.1.1 provides a powerful means to capture the decision-making of each individual user in the transportation network. But how can we use it to capture the collective behavior of a population of users, which is relevant in network evaluation and design?

To answer this question, consider a population, *i.e.*, a large amount, of users in a common transportation network. Each individual user is trying to solve a finite horizon MDP, where the stochasticity is due to imperfect modeling of the environment. The MDPs of different users will share same states, action, time steps and transition probability, since they are determined by the common network. However, the cost of choosing an action will increase with its user volumes, *i.e.*, amount of users choosing it, in the same sense that travel time on a highway increases with the amount of vehicles, or the price in the market increases with the amount of interested buyers. In this case, since the cost of individual user's action sequence depends on the collective behavior of all users, it is no longer meaningful to search for the “optimal strategy" for an individual user. Instead, we are interested in an *Wardrop equilibrium behavior* of a population game played by all network users, where no individual user can benefit from unilateral switching actions; in other words, every user is using an “optimal" action sequence.

We formally define the above problem as a population game in Problem 2.2, and group our assumptions in Assumption 2.2. Notice that we allow the amount of users to be continuous real numbers rather than integers. Such relaxation is common in applications where the amount of the user population is large enough: it is fairly reasonable to round-off 999.5 users to 1000 users, but much less reasonable to round-off 1.5 users to 2 users.

Problem 2.2. *Consider the following T-stage population game. At each time $t \in [T]$, p_{ts} new users start the game from state $s \in [S]$; each such player can choose to take action $a \in [A]$ at the cost of $\phi_{tsa}(y_{tsa})$, where y_{tsa} denote the amount of users choose to take action a in state s at time t, and reach state $s' \in [S]$ with probability $P_{sas'}$ at time $t+1$, then repeat such process till $t = T$, when the player end the game after choosing the last action. Find an equilibrium distribution of the users where no individual player can benefit from unilaterally switching action sequences.*

Assumption 2.2. *We assume $p \in \mathbb{R}_+^{T\times S}$, $P \in \mathbb{R}_+^{S\times A\times S}$ and $\sum_{s'} P_{sas'} = 1$ for all $s \in [S]$, $a \in [A]$. Further, function $\phi_{tsa} : [0, \rho] \to \mathbb{R}$ is continuous and strictly increasing, where $\rho = \sum_{t,s} p_{ts}$.*

Given the success of solving Problem 2.1 via linear programming, is it possible to solve Problem 2.2 using convex optimization too? Theorem 2.2 shows that the answer is affirmative by proving the connection between Problem 2.2 and the following two convex optimization problems, where

- variable y_{tsa} in (2.7) denotes the amount of users choosing action a in state s at time t
- variable u_{tsa} in (2.8) denotes the cost of choosing action a in state s at time t

$$
\begin{aligned}
\underset{y}{\text{minimize}} \quad & \sum_{t,s,a} \int_0^{y_{tsa}} \phi_{tsa}(\alpha) d\alpha \\
\text{subject to} \quad & \sum_a y_{1sa} = p_{1s}, \\
& \sum_a y_{t+1,sa} = p_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a}, \quad t \in [T-1], \\
& 0 \leqslant y_{tsa}, \quad \forall t \in [T], s \in [S], a \in [A].
\end{aligned} \tag{2.7}
$$

$$
\begin{aligned}
\underset{u,v}{\text{maximize}} \quad & \sum_{t,s} p_{ts} v_{ts} - \sum_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha \\
\text{subject to} \quad & v_{Ts} \leqslant u_{Tsa}, \\
& v_{ts} \leqslant u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}, \\
& \forall t \in [T-1], s \in [S], a \in [A].
\end{aligned} \tag{2.8}
$$

Theorem 2.2. *Suppose Assumption 2.2 holds, y solves* (2.7)*, and* (u, v) *solves* (2.8)*. Further, if $y_{tsa} > 0$, then*

$$
\begin{aligned}
(v_{Ts}, a) &= \min_{a' \in [A]} \ \phi_{Tsa'}(y_{Tsa'}), \\
(v_{ts}, a) &= \min_{a' \in [A]} \ \phi_{tsa'}(y_{tsa'}) + \sum_{s'} P_{sa's'} v_{t+1,s'},
\end{aligned} \tag{2.9}
$$

for all $t \in [T-1]$, where we use the convention in (2.3)*.*

Proof. The objective function of problem (2.7) is convex (since ϕ_{tsa} is strictly increasing), its constraints are affine, and the optimal value is obviously finite (since ϕ_{tsa} is finitely valued). These imply that a solution pair to (2.7) and (2.8) necessarily satisfy the KKT conditions [152, Thm. 28.3.1]. Let v_{ts} be the dual variable corresponding to the equality constraints containing p_{ts}, let $\mu_{tsa} \geqslant 0$ be the dual variables corresponding to constraint $y_{tsa} \geqslant 0$. Then the Lagrangien of (2.7) and (2.8) is given by

$$
L(y, v, \mu) = \sum_{t,s,a} \int_0^{y_{tsa}} \phi_{tsa}(\alpha) d\alpha - \sum_{t,s,a} \mu_{tsa} y_{tsa} + \sum_{t,s,a} v_{ts}(p_{ts} - y_{tsa}) + \sum_{t, t<T} \sum_{s',a,s} v_{t+1,s} P_{s'as} y_{ts'a}.
$$

The KKT conditions [152, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L/\partial y_{tsa}$ equal to zero)

$$\begin{aligned} v_{Ts} =& \phi_{Tsa}(y_{Tsa}) - \mu_{Tsa}, \\ v_{ts} =& \phi_{tsa}(y_{tsa}) + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa}, \ \ t \in [T-1], \end{aligned} \tag{2.10}$$

for all $s \in [S], a \in [A]$, and the complementarity conditions

$$y_{tsa}\mu_{tsa} = 0, \ \ y_{tsa} \geqslant 0, \ \ \mu_{tsa} \geqslant 0, \ \ \forall t \in [T], s \in [S], a \in [A] \tag{2.11}$$

Combining (2.10) and (2.11) gives the desired results. Note that same results can be derived from the dual problem (2.8). □

Theorem 2.2 shows that solutions to optimization (2.7) and (2.8) provides a Wardrop equilibrium of Problem 2.2 in the following sense. If action a is chosen by some users in state s at time t, *i.e.*, $y_{tsa} > 0$, then action a is optimal among the alternative actions in the sense of (2.9), which is identical to (2.4) where $c_{tsa} = \phi_{tsa}(y_{tsa})$ for all $t \in [T], s \in [S], a \in [A]$. In other words, every user is choosing the optimal actions given the prevailing choice of other users, and no individual users can benefit from unilaterally switching policies.

So how can we solve optimization (2.7) and (2.8)? Unfortunately, their KKT conditions cannot be solved directly using dynamic programming like those of (2.1) and (2.2), due to the nonlinear terms in (2.10). However, this does not render Algorithm 1 and Algorithm 2 completely useless in solving (2.7) and (2.8). For example, notice that if the objective function in (2.7) is linearized around a reference point, then it reduces to an instance of linear programming (2.1). Hence we can solve (2.7) using the conditional gradient method (see Appendix B.5), which solves (2.7) by recursively taking convex combination of solutions to instances of (2.1). Similarly, notice that (2.8) can be rewritten as follows

$$\underset{u}{\text{maximize}} \quad g(u) - \sum_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha \tag{2.12}$$

where

$$\begin{aligned} g(u) = \underset{v}{\text{maximize}} \quad & \sum_{t,s} p_{ts} v_{ts} \\ \text{subject to} \quad & v_{Ts} \leqslant u_{Tsa} \\ & v_{ts} \leqslant u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}, \\ & \forall t \in [T-1], s \in [S], a \in [A]. \end{aligned} \tag{2.13}$$

Using the strong duality of linear programming [169], the above equation is equivalent to

$$\begin{aligned} g(u) = \underset{y}{\text{minimize}} \quad & \sum_{t,s,a} u_{tsa} y_{tsa} \\ \text{subject to} \quad & \sum_{a} y_{1sa} = p_{1s}, \\ & \sum_{a} y_{t+1,sa} = p_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a}, \;\; t \in [T-1], \\ & y_{tsa} \geqslant 0, \;\; \forall t \in [T], s \in [S], a \in [A] \end{aligned} \tag{2.14}$$

In other words, $g(u)$ is the support function of a polytope defined by the constraints in (2.14), whose subgradients are given by the optimizers in linear programming (2.14); see [152, Cor. 23.5.3]. Therefore we can apply projected subgradient method; see Appendix B.2. We summarized the conditional gradient method for (2.14) and projected subgradient method for (2.13) in Algorithm 3 and, respectively, Algorithm 4, where we adopt the following notations for compactness

$$\begin{aligned} & \phi(y), \phi^{-1}(u) \in \mathbb{R}^{T\times S\times A}, \;\; [\phi(y)]_{tsa} = \phi_{tsa}(y_{tsa}), \;\; [\phi^{-1}(u)]_{tsa} = \phi^{-1}_{tsa}(u_{tsa}), \\ & \underline{u}_{tsa} = \phi_{tsa}(0), \;\; \overline{u}_{tsa} = \phi_{tsa}(\rho). \end{aligned} \tag{2.15}$$

2.2 Variations

In this section, we discuss two variations of the Markovian network equilibrium model introduced in §2.1.2. Both variations come from classical network equilibrium theory. Again, our main goal is to show how these theories are modified due to the change from shortest path

Algorithm 3 Conditional gradient method

Input: $p, P, \phi, \psi, T, \{\alpha^k\}$, initial value for y, z.

for $k = 1, 2, \ldots, K$ **do**
  $(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1$(P, \phi(y), T)$.
  $\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p, P, T)$.
  $y \leftarrow y - \frac{2}{k+1}(y - \hat{y})$
end for

Algorithm 4 Projected subgradient method

Input: $P, p, \phi, \psi, T, \{\alpha^k\}$, initial value for u, w.

for $k = 1, 2, \ldots, K$ **do**
  $(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1(P, u, T)
  $\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p - \hat{z}, P, T)$.
  $u \leftarrow \min\{\overline{u}, \max\{\underline{u}, u + \alpha^k\,(\hat{y} - \phi^{-1}(u))\}\}$
end for

problem to MDP.

2.2.1 *Variable demand*

One of the limitation of network equilibrium model introduced in §2.1.2 is that the total amount of users is fixed. For example, in the transportation network example we introduced at the beginning of the chapter, it is possible that some users will cancel their plans if the cost of using the network is too high. In this case, one can intuitively think the network equilibrium as a market model. The supply side is the network itself, providing the users the options of making sequence of decision. The demand of the market corresponds to the amount of users in the network, which changes with the expected accumulated cost of using the network. The model in §2.1.2 only accounts for the supply side of the story. How can we augment it to account for the demand side as well?

With this motivation question in mind, we propose the following extension to Problem 2.2, which allows the actual amount of users in the network to vary by including a quitting option; we group our assumptions in Assumption 2.3.

Problem 2.3. *Consider the following T-stage population game. At each time $t \in [T]$, p_{ts} new users start the game from state $s \in [S]$; each such user can choose to*

1. *quit the game immediately at the cost of $\psi_{ts}(z_{ts})$*

2. *take action $a \in [A]$ at the cost of $\phi_{tsa}(y_{tsa})$ and reach state $s' \in [S]$ with probability $P_{sas'}$ at time $t+1$, then repeat such process till $t = T$, when the user end the game after choosing the last action*

where z_{ts} and y_{tsa} denote the total amount of users choose to quit the game and, respectively, taking action a in state s at time t. Find an equilibrium distribution of the users where no individual user can benefit from unilaterally switching actions.

Assumption 2.3. *We assume $p \in \mathbb{R}_+^{T\times S}$, $P \in \mathbb{R}_+^{S\times A\times S}$ and $\sum_{s'} P_{sas'} = 1$ for all $s \in [S]$, $a \in [A]$. Further, function $\phi_{tsa} : [0,\rho] \to \mathbb{R}$ and $\psi_{ts} : [0,\rho] \to \mathbb{R}$ are continuous and strictly increasing over their respective domain, where $\rho = \sum_{t,s} p_{ts}$.*

Similar to Theorem 2.2, Theorem 2.3 shows that Problem 2.3 can also be solved via convex optimization problems, given as follows.

$$
\begin{array}{ll}
\underset{y,z}{\text{minimize}} & \sum\limits_{t,s,a} \int_0^{y_{tsa}} \phi_{tsa}(\alpha) d\alpha + \sum\limits_{t,s} \int_0^{z_{ts}} \psi_{ts}(\alpha) d\alpha \\
\text{subject to} & \sum\limits_a y_{1sa} = p_{1s} - z_{1s}, \\
& \sum\limits_a y_{t+1,sa} = p_{t+1,s} - z_{t+1,s} + \sum\limits_{s',a} P_{s'as} y_{ts'a}, \ \ t \in [T-1], \\
& 0 \leqslant y_{tsa}, 0 \leqslant z_{ts} \leqslant p_{ts}, \ \forall t \in [T], s \in [S], a \in [A].
\end{array}
\tag{2.16}
$$

$$
\begin{array}{ll}
\underset{u,v,w,\lambda}{\text{maximize}} & \sum\limits_{t,s} p_{ts}(v_{ts} - \lambda_{ts}) - \sum\limits_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha - \sum\limits_{t,s} \int_{\psi_{ts}(0)}^{w_{ts}} \psi_{ts}^{-1}(\alpha) d\alpha \\
\text{subject to} & v_{Ts} \leqslant u_{Tsa}, \\
& v_{ts} \leqslant u_{tsa} + \sum\limits_{s'} P_{sas'} v_{t+1,s'}, \ \ t \in [T-1], \\
& v_{ts} \leqslant w_{ts} + \lambda_{ts}, \ \ 0 \leqslant \lambda_{ts}, \\
& \forall t \in [T], s \in [S], a \in [A].
\end{array}
\tag{2.17}
$$

Theorem 2.3. *Suppose Assumption 2.3 holds, (y, z) solves (2.16), and (u, v, w, λ) solves (2.17). For any $p_{ts} > 0$,*

$$\begin{aligned} &\text{if } z_{ts} = 0, \ \ \text{then } v_{ts} \leqslant \psi_{ts}(p_{ts}),\\ &\text{if } 0 < z_{ts} < p_{ts}, \ \ \text{then } v_{ts} = \psi_{ts}(z_{ts}),\\ &\text{if } z_{ts} = p_{ts}, \ \ \text{then } v_{ts} \geqslant \psi_{ts}(0). \end{aligned} \tag{2.18}$$

Further, if $y_{tsa} > 0$, then

$$\begin{aligned} (v_{Ts}, a) &= \min_{a' \in [A]} \ \phi_{Tsa'}(y_{Tsa'}),\\ (v_{ts}, a) &= \min_{a' \in [A]} \ \phi_{tsa'}(y_{tsa'}) + \sum_{s'} P_{sa's'} v_{t+1,s'}, \end{aligned} \tag{2.19}$$

for all $t \in [T-1]$, where we use the convention in (2.3).

Proof. The objective function of problem (2.16) is convex (since ϕ_{tsa} and ψ_{ts} are strictly increasing), its constraints are affine, and the optimal value is obviously finite (since ϕ_{tsa} and ψ_{ts} are finitely valued). These imply that a solution pair to (2.16) and (2.17) necessarily satisfy the KKT conditions [152, Thm. 28.3.1]. Let v_{ts} be the dual variable corresponding to the equality constraints containing p_{ts}, let $\mu_{tsa}, \theta_{ts}, \lambda_{ts} \geqslant 0$ be the dual variables corresponding to constraint $y_{tsa} \geqslant 0$, $z_{ts} \geqslant 0$ and, respectively, $z_{ts} \leqslant p_{ts}$. Then the Lagrangien of (2.16) and (2.17) is given by

$$\begin{aligned} L(y, z, v, \mu, \lambda, \theta) = &\sum_{t,s,a} \int_0^{y_{tsa}} \phi_{tsa}(\alpha) d\alpha - \sum_{t,s,a} \mu_{tsa} y_{tsa} + \sum_{t,s} \int_0^{z_{ts}} \psi_{ts}(\alpha) d\alpha\\ &+ \sum_{t,s,a} v_{ts}(p_{ts} - z_{ts} - y_{tsa}) + \sum_{t,t<T} \sum_{s',a,s} v_{t+1,s} P_{s'as} y_{ts'a} - \sum_{t,s} ((\theta_{ts} - \lambda_{ts}) z_{ts} + \lambda_{ts} p_{ts}) \end{aligned}$$

The KKT conditions [152, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L/\partial y_{tsa}, \partial L/\partial z_{ts}$ equal to zero)

$$
\begin{aligned}
v_{Ts} =& \phi_{Tsa}(y_{Tsa}) - \mu_{Tsa}, \\
v_{ts} =& \phi_{tsa}(y_{tsa}) + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa}, \;\; t \in [T-1], \\
v_{ts} =& \psi_{ts}(z_{ts}) + \lambda_{ts} - \theta_{ts}, \;\; t \in [T],
\end{aligned}
\tag{2.20}
$$

for all $s \in [S], a \in [A]$, and the complementarity conditions

$$
\begin{aligned}
& y_{tsa}\mu_{tsa} = 0, \;\; z_{ts}\theta_{ts} = 0, \;\; \lambda_{ts}(z_{ts} - p_{ts}) = 0 \\
& y_{tsa} \geqslant 0, \, z_{ts} \geqslant 0, \, \mu_{tsa} \geqslant 0, \, \theta_{ts} \geqslant 0, \, \lambda_{ts} \geqslant 0, \;\; \forall t \in [T], s \in [S], a \in [A]
\end{aligned}
\tag{2.21}
$$

Combining (2.20) and (2.21) yields (2.18) and (2.19). Note that same results can be derived from the dual problem (2.17). □

If we fix the variable z in (2.16) to take value zero, then it reduces to (2.7). This suggest that a modification of Algorithm 4 can be used to solve (2.16). To see this, we first write problem (2.16) with linearized objective as follows

$$
\begin{aligned}
g(u, w) = \underset{y,z}{\text{minimize}} \quad & \sum_{t,s,a} u_{tsa} y_{tsa} + \sum_{t,s} w_{ts} z_{ts} \\
\text{subject to} \quad & \sum_a y_{1sa} = p_{1s} - z_{1s}, \\
& \sum_a y_{t+1,sa} = p_{t+1,s} - z_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a}, \;\; t \in [T-1], \\
& 0 \leqslant y_{tsa}, 0 \leqslant z_{ts} \leqslant p_{ts}, \, \forall t \in [T], s \in [S], a \in [A].
\end{aligned}
\tag{2.22}
$$

where $u \in \mathbb{R}^{T \times S \times A}$ and $w \in \mathbb{R}^{T \times S}$ are linearization constants. The following lemma shows an optimizer to (2.22) can be obtained via Algorithm 1 and Algorithm 2.

Lemma 2.4. *Suppose Assumption 2.3 holds. Let $(\hat{v}, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, T), $\hat{z} = p \odot (\hat{v} > w)$, $\hat{y}$ be the output of Algorithm 2 with input $(\hat{\pi}, p - \hat{z}, P, T)$.*

Then

$$g(u,w) = \sum_{t,s,a} u_{tsa}\hat{y}_{tsa} + \sum_{t,s} w_{ts}\hat{z}_{ts}.$$

Proof. Using a similar argument as in the proof of Theorem 2.3, we can show that the KKT conditions of (2.22) are given by the following

$$\begin{aligned}
\sum_a y_{1sa} =& p_{1s} - z_{1s} \\
\sum_a y_{t+1,sa} =& p_{t+1,s} - z_{t+1,s} + \sum_{s',a} P_{s'as} y_{ts'a},\ t \in [T-1], \\
v_{Ts} =& u_{Tsa} - \mu_{Tsa}, \\
v_{ts} =& u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'} - \mu_{tsa},\ \ t \in [T-1], \\
v_{ts} =& w_{ts} + \lambda_{ts} - \theta_{ts}, \\
y_{tsa}\mu_{tsa} =& 0,\ \ z_{ts}\theta_{ts} = 0,\ \ \lambda_{ts}(z_{ts} - p_{ts}) = 0 \\
y_{tsa},\ z_{ts}, & \mu_{tsa},\ \theta_{ts},\ \lambda_{ts} \geqslant 0,
\end{aligned} \tag{2.23}$$

for all $t \in [T], s \in [S], a \in [A]$. Let $(\hat{v}, \hat{\pi})$ be the output of Algorithm 1 with input (P, u, T), $\hat{z} = p \odot (\hat{v} > w)$, $\hat{y}$ be the output of Algorithm 2 with input $(\hat{\pi}, p - \hat{z}, P, T)$, let

$$\begin{aligned}
\hat{\mu}_{Tsa} &= -\hat{v}_{Ts} + u_{Tsa}, \\
\hat{\mu}_{tsa} &= -\hat{v}_{ts} + u_{tsa} + \sum_{s'} P_{sas'}\hat{v}_{t+1,s'},\ \ t \in [T-1], \\
\hat{\lambda}_{ts} &= \max\{\hat{v}_{ts} - w_{ts}, 0\},\ \ \hat{\theta}_{ts} = \max\{w_{ts} - \hat{v}_{ts}, 0\}
\end{aligned} \tag{2.24}$$

for all $t \in [T], s \in [S], a \in [A]$. Then it is straightforward to verify that $(\hat{y}, \hat{z}, \hat{v}, \hat{\mu}, \hat{\lambda}, \hat{\theta})$ satisfies all the KKT conditions in (2.23), hence $(\hat{y}, \hat{z})$ solves (2.22), which completes the proof. □

Lemma 2.3 shows that when its objective function is linearized, optimization (2.16) can be solved by dynamic programming, just like the one in (2.14). Based on this observation, we propose to Algorithm 5, a conditional gradient method designed for optimization (2.29), where we will use the following notation

$$z = p \odot (v > w) \iff z_{ts} = \begin{cases} p_{ts}, & v_{ts} > w_{ts} \\ 0, & v_{ts} \leqslant w_{ts} \end{cases}, \quad \forall t \in [T], s \in [S]. \tag{2.25}$$

What about the optimization in (2.17)? Similar to the one in (2.8), we can rewrite (2.17) as follows

$$\underset{u,w}{\text{maximize}} \quad g(u,w) - \sum_{t,s,a} \int_{\phi_{tsa}(0)}^{u_{tsa}} \phi_{tsa}^{-1}(\alpha) d\alpha - \sum_{t,s} \int_{\psi_{ts}(0)}^{w_{ts}} \psi_{ts}^{-1}(\alpha) d\alpha \tag{2.26}$$

where

$$\begin{aligned} g(u,w) = \underset{v,\lambda}{\text{maximize}} \quad & \sum_{t,s} p_{ts}(v_{ts} - \lambda_{ts}) \\ \text{subject to} \quad & v_{Ts} \leqslant u_{Tsa}, \\ & v_{ts} \leqslant u_{tsa} + \sum_{s'} P_{sas'} v_{t+1,s'}, \;\; t \in [T-1], \\ & v_{ts} \leqslant w_{ts} + \lambda_{ts}, \;\; 0 \leqslant \lambda_{ts}, \\ & \forall t \in [T], s \in [S], a \in [A]. \end{aligned} \tag{2.27}$$

Again, using strong duality of linear programming, one can verify that the definition of function $g(u,w)$ in (2.22) agrees with the one in (2.27), and an optimizer to the optimization in (2.22) gives a subgradient to function $g(u,w)$; see [152, Cor. 23.5.3]. Based on this observation, we propose to solve the optimization in (2.17) using the projected subgradient method summarized in Algorithm 6, where we also use the following notation

$$\underline{u}_{tsa} = \phi_{tsa}(0), \;\; \overline{u}_{tsa} = \phi_{tsa}(\rho), \;\; \underline{w}_{ts} = \psi_{ts}(0), \;\; \overline{w}_{ts} = \psi_{ts}(\rho), \tag{2.28}$$

for all $t \in [T], s \in [S], a \in [A]$, where ρ is defined as in Assumption 2.3.

Algorithm 5 Frank-Wolfe method

Input: $p, P, \phi, \psi, T, \{\alpha^k\}$, initial value for y, z.

for $k = 1, 2, \ldots, K$ **do**
    $(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1$(P, \phi(y), T)$.
    $\hat{z} \leftarrow p \odot (\hat{v} > \psi(z))$ ▷ cf. (2.25)
    $\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p - \hat{z}, P, T)$.
    $y \leftarrow y - \frac{2}{k+1}(y - \hat{y})$
    $z \leftarrow z - \frac{2}{k+1}(z - \hat{z})$
end for

Algorithm 6 Subgradient method

Input: $P, p, \phi, \psi, T, \{\alpha^k\}$, initial value for u, w.

for $k = 1, 2, \ldots, K$ **do**
    $(\hat{v}, \hat{\pi}) \leftarrow$ Alg. 1(P, u, T)
    $\hat{z} \leftarrow p \odot (\hat{v} > w)$ ▷ cf.(2.25)
    $\hat{y} \leftarrow$ Alg. 2 $(\hat{\pi}, p - \hat{z}, P, T)$.
    $u \leftarrow \min\{\overline{u}, \max\{\underline{u}, u + \alpha^k\left(\hat{y} - \phi^{-1}(u)\right)\}\}$
    $w \leftarrow \min\{\overline{w}, \max\{\underline{w}, w + \alpha^k\left(\hat{z} - \psi^{-1}(w)\right)\}\}$ ▷ cf.(2.28)
end for

2.2.2 Multi-commodity flow

Another limitation of the network equilibrium model in §2.1.2 is that all users have the same planning horizon T. In practice, it is more likely that users with heterogeneous planning horizons will use the network facility simultaneously. How should the model in §2.1.2 be adjusted to account for this scenarios? The current section is devoted to answering this question.

We start with the following generalization to Problem 2.2, which allows users with heterogeneous ending time; we group our assumptions in Assumption 2.4. For simplicity here we only consider fixed demand scenario; an extension that combines the variable demand feature in §2.2.1 is straightforward.

Problem 2.4. *Consider the following T-stage population game. At each time $t \in [T]$, for each ending time $\tau \in \mathbb{T} \subset [T]$ such that $\tau \geqslant t$, p^{τ}_{ts} new players start the game from state s.*

Each such player can choose action a at the cost of $\phi_{tsa}(\sum_{\tau,\tau\geqslant t} y^{\tau}_{tsa})$ and reach state s' with probability $P_{sas'}$ at time $t+1$, then repeat this process till $t=\tau$, when the player end the game after choosing the last action. Here y^{τ}_{tsa} denote the total amount of players who plan to end the game at time τ and choose action a in state s at time t. Find an equilibrium distribution of the players where no individual player can benefit from unilaterally switching actions.

Assumption 2.4. *We assume $T \in \mathbb{T} \subseteq [T]$, $p^{\tau}_{ts} \in \mathbb{R}_+$ for all $\tau \in \mathbb{T}$, $t \geqslant \tau$ and $s \in [S]$; $P \in \mathbb{R}^{S\times A\times S}_+$ and $\sum_{s'} P_{sas'} = 1$ for all $s \in [S]$, $a \in [A]$. Further function $\phi_{tsa} : [0,\rho] \to \mathbb{R}$ is continuous and strictly increasing, where $\rho = \sum_{\tau}\sum_{t\leqslant\tau,s} p^{\tau}_{ts}$.*

Notice that Problem 2.2 is a special case of Problem 2.4 with $\mathbb{T} = \{T\}$. Based on this observation, we propose the following two optimization problems that reduces to (2.7) and, respectively, (2.13) if $\mathbb{T} = \{T\}$. In addition, Theorem 2.5 shows that how the solution to these optimization problems solve Problem 2.4, in the same sense as Theorem 2.2.

$$
\begin{aligned}
\underset{y^{\tau},\tau\in\mathbb{T}}{\text{minimize}} \quad & \sum_{t,s,a}\int_0^{\sum\limits_{\tau,\tau\geqslant t} y^{\tau}_{tsa}} \phi_{tsa}(\alpha)d\alpha \\
\text{subject to} \quad & \sum_a y^{\tau}_{1sa} = p^{\tau}_{1s} \\
& \sum_a y^{\tau}_{t+1,sa} = p^{\tau}_{t+1,s} + \sum_{s',a} P_{s'as} y^{\tau}_{ts'a},\ t \in [\tau-1], \\
& 0 \leqslant y^{\tau}_{tsa},\quad \forall \tau \in \mathbb{T}, t\in[\tau], s\in[S], a\in[A]
\end{aligned}
\tag{2.29}
$$

$$
\begin{aligned}
\underset{u,v^{\tau},\tau\in\mathbb{T}}{\text{maximize}} \quad & \sum_{t,s}\sum_{\tau,\tau\geqslant t} p^{\tau}_{ts} v^{\tau}_{ts} - \sum_{t,s,a}\int_{\phi_{tsa}(0)}^{u_{tsa}} \phi^{-1}_{tsa}(\alpha)d\alpha \\
\text{subject to} \quad & v^{\tau}_{\tau s} \leqslant u_{\tau sa}, \\
& v^{\tau}_{ts} \leqslant u_{tsa} + \sum_{s'} P_{sas'} v^{\tau}_{t+1,s'}, \\
& \forall \tau \in \mathbb{T}, t\in[\tau-1], s\in[S], a\in[A]
\end{aligned}
\tag{2.30}
$$

Theorem 2.5. *Suppose Assumption 2.4 holds, y solves (2.29), and (u,v) solves (2.30). If*

$y^{\tau}_{tsa} > 0$ *for any* $\tau \in \mathbb{T}, s \in [S], a \in [A]$, *then*

$$\begin{aligned}
(v^{\tau}_{\tau s}, a) &= \min_{a' \in [A]} \ \phi_{\tau s a'}\Big(\sum_{\tau, \tau \geqslant t} y^{\tau}_{\tau s a'} \Big), \\
(v^{\tau}_{ts}, a) &= \min_{a' \in [A]} \ \phi_{tsa'}\Big(\sum_{\tau, \tau \geqslant t} y^{\tau}_{tsa'} \Big) + \sum_{s'} P_{sa's'} v^{\tau}_{t+1, s'},
\end{aligned} \tag{2.31}$$

for all $t \in [\tau - 1]$, *where we use the convention in* (2.3).

Proof. The objective function of problem (2.29) is convex (since ϕ_{tsa} is increasing), its constraints are affine, and the optimal value is obviously finite (since ϕ_{tsa} is finitely valued).These imply that a solution pair to (2.29) and (2.30) necessarily satisfy the KKT conditions [152, Cor. 28.3.1]. Let v^{τ}_{ts} be the dual variable corresponding to the equality constraints containing $p^{\tau}_{t}s$, let $\mu^{\tau}_{tsa} \geqslant 0$ be the dual variables corresponding to constraint $y^{\tau}_{tsa} \geqslant 0$. Then the Lagrangien of (2.29) and (2.30) is given by

$$\begin{aligned}
L(y, v, \mu) = &\sum_{t,s,a} \int_{0}^{\sum_{\tau, \tau \geqslant t} y^{\tau}_{tsa}} \phi_{tsa}(\alpha) d\alpha + \sum_{\tau} \sum_{t, t \leqslant \tau} \sum_{s,a} (v^{\tau}_{ts}(p^{\tau}_{ts} - y^{\tau}_{tsa}) - \mu^{\tau}_{tsa} y^{\tau}_{tsa}) \\
&+ \sum_{\tau} \sum_{t, t < \tau} \sum_{s', a, s} v^{\tau}_{t+1, s} P_{s'as} y^{\tau}_{ts'a}.
\end{aligned}$$

The KKT conditions [152, Thm.28.3] of this Lagrangian include the following vanishing gradient conditions (by setting $\partial L / \partial y^{\tau}_{tsa}$ equal to zero)

$$\begin{aligned}
v^{\tau}_{\tau s} = &\phi_{\tau s a}\Big(\sum_{j, j \geqslant \tau} y^{j}_{\tau s a} \Big) - \mu^{\tau}_{\tau s a} = 0, \\
v^{\tau}_{ts} = &\phi_{tsa}\Big(\sum_{j, j \geqslant t} y^{j}_{tsa} \Big) + \sum_{s'} P_{sas'} v^{\tau}_{t+1, s'} - \mu^{\tau}_{tsa} = 0,
\end{aligned} \tag{2.32}$$

for all $t \in [\tau - 1], \tau \in \mathbb{T}, s \in [S], a \in [A]$, and the complementarity conditions

$$y^{\tau}_{tsa} \mu^{\tau}_{tsa} = 0, \ \ y^{\tau}_{tsa} \geqslant 0, \ \mu^{\tau}_{tsa} \geqslant 0, \tag{2.33}$$

for all $t \in [\tau], \tau \in \mathbb{T}, s \in [S], a \in [A]$. Combining (2.32) and (2.33) yields (2.31). Again, same results can be derived from the dual problem (2.30). ☐

Similarly, if we approximate the objective function in (2.29) using a linear function, we obtain the following

$$\begin{aligned} h(u) = \underset{y^\tau, \tau \in \mathbb{T}}{\text{minimize}} \quad & \sum_{t,s,a} \sum_{\tau, \tau \geqslant t} u_{tsa} y^\tau_{tsa} \\ \text{subject to} \quad & \sum_a y^\tau_{1sa} = p^\tau_{1s} \\ & \sum_a y^\tau_{t+1,sa} = p^\tau_{t+1,s} + \sum_{s',a} P_{s'as} y^\tau_{ts'a}, \; t \in [\tau - 1], \\ & 0 \leqslant y^\tau_{tsa}, \quad \forall \tau \in \mathbb{T}, t \in [\tau], s \in [S], a \in [A] \end{aligned} \tag{2.34}$$

where $u \in \mathbb{R}^{T \times S \times A}$ is the approximation parameter, $-h(u)$ is the optimal value of (2.34)

Lemma 2.6. *Suppose Assumption 2.4 holds. Let* $(\hat{v}^\tau, \hat{\pi})$ *be the output of Algorithm 1 with input* (P, u, τ)*,* $\hat{y}^\tau$ *be the output of Algorithm 2 with input* $(\hat{\pi}^\tau, p^\tau, P, \tau)$*. Then*

$$h(u) = \sum_{t,s,a} \sum_{\tau, \tau \geqslant t} u_{tsa} \hat{y}^\tau_{tsa}.$$

Proof. Notice that in optimization (2.34), both objective function and constraints are completely separable across y^τ with different value of τ. In other words, solving (2.34) is equivalent to solve the following optimization problem for each value of $\tau \in \mathbb{T}$ separately

$$\begin{aligned} \underset{y^\tau}{\text{minimize}} \quad & \sum_{t \leqslant \tau, s, a} u_{tsa} y^\tau_{tsa} \\ \text{subject to} \quad & \sum_a y^\tau_{1sa} = p^\tau_{1s} \\ & \sum_a y^\tau_{t+1,sa} = p^\tau_{t+1,s} + \sum_{s',a} P_{s'as} y^\tau_{ts'a}, \quad t \in [\tau - 1], \\ & 0 \leqslant y^\tau_{tsa}, \quad \forall t \in [\tau], s \in [S], a \in [A] \end{aligned} \tag{2.35}$$

Since problem (2.35) is nothing but an instance of (2.1) with $T = \tau$, it can be solved by the output of Algorithm 2 with input $(\hat{\pi}^\tau, p^\tau, P, \tau)$, where $\hat{\pi}^\tau$ is the output of Algorithm 1 with input (P, u, τ). This completes the proof. ☐

Using Lemma 2.6 and arguments similar to those in §2.2.1, we propose to solve optimization (2.29) and (2.30) using the conditional gradient method in Algorithm 7 and, respectively, projected subgradient method in Algorithm 8, where we use the following short-hand notation

$$y = \oplus y^\tau \iff y_{tsa} = \textstyle\sum_{\tau, \tau \geqslant t} y^\tau_{tsa}, \quad \forall t \in [T], s \in [S], a \in [A]. \tag{2.36}$$

Algorithm 7 Multicommodity conditional gradient method

Input: $p, P, \phi, \mathbb{T}, \{\alpha^k\}$, initial value for y^τ for all $\tau \in \mathbb{T}$.

for $k = 1, 2, \ldots, K$ **do**
- $\hat{y} \leftarrow \oplus y^\tau$ ▷ cf. (2.36)
- $(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow$Alg. $1(P, \phi(\hat{y}), \tau), \ \ \forall \tau \in \mathbb{T}$
- $\hat{y}^\tau \leftarrow$ Alg. $2(\hat{\pi}^\tau, p^\tau, P, \tau), \ \ \forall \tau \in \mathbb{T}$
- $y^\tau \leftarrow y^\tau - \frac{2}{k+1}(y^\tau - \hat{y}^\tau), \ \ \forall \tau \in \mathbb{T}$

end for

Algorithm 8 Multicommodity subgradient method

Input: $p, P, \phi, \mathbb{T}, \{\alpha^k\}$, initial vaue of u.

for $k = 1, 2, \ldots, K$ **do**
- $(\hat{v}^\tau, \hat{\pi}^\tau) \leftarrow$ Alg. $1(P, u, \tau), \ \ \forall \tau \in \mathbb{T}$
- $\hat{y}^\tau \leftarrow$Alg. $2(\hat{\pi}^\tau, p^\tau, P, \tau), \ \ \forall \tau \in \mathbb{T}$
- $\hat{y} \leftarrow \oplus \hat{y}^\tau$ ▷ cf. (2.36)
- $u \leftarrow \min\{\overline{u}, \max\{\underline{u}, u + \alpha^k \left(\hat{y} - \phi^{-1}(u)\right)\}\}$

end for

2.3 Numerical examples

In this section, we first demonstrate the theoretical results of this chapter with a multicommodity ridesharing example in §2.3.1, then demonstrate the computational advantages of Frank-Wolfe and projected subgradient method in solving Markovian network equilibrium problems against off-the-shelf convex optimization software.

2.3.1 Multicommodity ridesharing game

We consider the game played by ridesharing drivers in Seattle competing for customers. We first abstract Seattle area as an undirected graph illustrated in Fig. 2.1, whose nodes denote various neighborhoods in Seattle, and edges denote available routes labeled by its driving distance. We denote $\mathcal{N}_s$ the set of neighboring nodes of node s. We model the decision making of an ridesharing driver on a typical weekend night (7pm-1am) as an MDP defined by the following.

- Time steps: $t = 1, 2, \ldots, 36$ denotes the (end of) 10-minute-intervals between 7pm and 1am.

- States: $[S]$ correspond to different nodes in graph $\mathcal{G}$.

- Actions: in state s, $a_{s'}$ denotes picking up an awaiting rider with destination s' for all $s' \in \mathcal{N}_s$; a_{wait} denotes waiting for a future rider.

- Transition kernel: we assume $P_{sas'}$ is given by

$$P_{sas'} = \begin{cases} 1 & \text{if } a = a_{s'},\ s' \in \mathcal{N}_s \\ 1/(|\mathcal{N}_s| + 1) & \text{if } a = a_{\text{wait}},\ s' \in \mathcal{N}_s \cup \{s\} \end{cases}$$

  All other entries of $P_{sas'}$ are zero. Here we use an uniform distribution over neighboring states to describe the uncertain destinations of future riders.[2]

- Cost: due to the competition among drivers, we assume the profit for picking up a rider decreases with the amount of drivers making the same offer, namely

$$f_{tss'} = \alpha + \beta\big(1 - \frac{y_{tsa_{s'}}}{\gamma_{tss'}}\big)\text{dist}_{ss'} \tag{2.37}$$

[2]Such distribution can be approximated more accurately using historical data in practical applications.

for all $t \in [T], s \in [S]$, where α and β is the baseline profit and, respectively, nominal profit per mile, $\text{dist}_{ss'}$ is the distance(miles) between s and s', $\gamma_{tss'}$ measures the rider demand from s to s' at time t, and finally $y_{tsa_{s'}}$ is the amount of drivers choosing action $a_{s'}$ in state s at time t. The cost of action a in state s is a function of y_{tsa} defined as follows

$$\phi_{tsa}(y_{tsa}) = \begin{cases} -f_{tss'} & \text{if } a = a_{s'}, s' \in \mathcal{N}_s \\ -\sum\limits_{s' \in \mathcal{N}_s} P_{sas'} f_{tss'} & \text{if } a = a_{\text{wait}} \end{cases}$$

- Planning time windows: We assume that 10 drivers start working from each state every 10 minutes from 7pm till 9pm. Each driver is assumed to only work for 4 consecutive hours to avoid driver fatigue, *i.e.*, $p_{ts}^{(t+24)} = 10$ for all $s \in [S]$ and $t = 1, 2, \ldots, 12$.

Notice that since drivers can start the game at different time during $1 \leqslant t \leqslant 12$ and they will only plan for the next 24 time steps. In other words, drivers with heterogeneous planning time windows will coexist in the network for $t = 2, 3, \ldots, 24$. Therefore the equilibrium of this game can be modeled as a multicommodity Markovian network equilibrium. We simulate the scenario where $\alpha = 10, \beta = 0.2$, $\gamma_{tss'}$ is given in Table 2.1 and $\text{dist}_{ss'}$ is given in Fig. 2.1. We demonstrate the driver number at downtown area $\mathcal{D} = \{9, 10, 11\}$ via Fig. 2.3, where we can see that driver number increase during $1 \leqslant t \leqslant 12$, then decrease during $24 \leqslant t \leqslant 36$. There are also two sudden changes in the increasing/decreasing speed around $t = 7$ and $t = 31$, which is due to the corresponding changes in values of γ in Table 2.1. This example extends the single commodity case considered in [30] where all players enter and exit the game simultaneously.

Further, our model can also be used for transportation network design. For example, suppose that Seattle city is considering two candidate light rail transit (LRT) routes, 7-9-10-11 and 6-8-9-11 (see Fig. 2.1), as means to alleviate the congestion caused by the ridesharing driver in downtown area, assuming LRT will reduce the demand of ridesharing service (namely, value of $\gamma_{tss'}$) by 50% along its route. The simulated equilibrium with different LRT routes are also given in Fig. 2.3, which shows that route 6-8-9-11 is more effective that route 7-9-10-11 in terms of reducing amount of drivers in $\mathcal{D}$. These results clearly demonstrate the power of Markovian network equilibrium model in transportation system design.

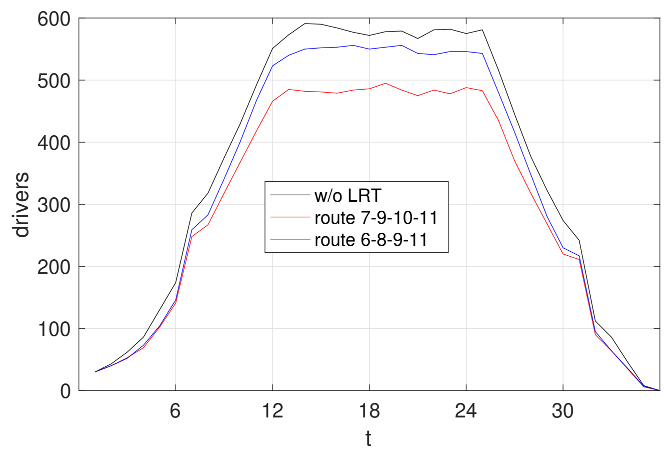


Figure 2.3: Driver numbers in downtown area $\mathcal{D} = \{9, 10, 11\}$.

Table 2.1: Values of γ where $\mathcal{D} = \{9, 10, 11\}$.

$\gamma_{tss'}$	$s \notin \mathcal{D},\ s' \in \mathcal{D}$	$s \in \mathcal{D},\ s' \in \mathcal{D}$	$s \in \mathcal{D},\ s' \notin \mathcal{D}$	$s \notin \mathcal{D},\ s' \notin \mathcal{D}$
$1 \leqslant t \leqslant 6$	600	200	60	60
$7 \leqslant t \leqslant 30$	200	400	200	60
$31 \leqslant t \leqslant 36$	60	100	600	60

2.3.2 *Computation experiments*

- *basic parameters*: let $P_{sas'} = \text{rand}(0, 1)^3$, then normalize it such that $\sum_{s'} P_{sas'} = 1$ for all $s \in [S], a \in [A]$;

- *cost functions*: let $\phi_{tsa}(y_{tsa}) = \text{rand}(1, 2)y_{tsa} + \text{rand}(1, 2)$ and $\psi_{ts}(x_{ts}) = \text{rand}(-110, -100)x_{ts} + \text{rand}(100, 110)$ for all $t \in [T], s \in [S], a \in [A]$.

[3] We denote $\text{rand}(a, b)$ a random number sampled from uniform distribution over interval $[a, b]$.

- *divergence*: let $p_{ts} = \text{rand}(0,1)$ for all $s \in [S]$ if $t = 1$ and zero otherwise.

We fix $T = A = 10$ and let S takes values in $\{50, 100, 150, 200, 250, 300, 350, 400\}$, and test the computation time of Algorithm 7 Algorithm 8, where both algorithms terminate when their objective function value agrees with the optimal one obtained by Gurobi with less than 10^{-4} relative error. The average computation time over 1000 examples, along with corresponding 3-standard deviation interval are reported in Fig. 2.4. All codes are in MATLAB and run on a 2.8GHz PC. From results in Fig. 2.4 we can see that, over the randomly generated 8000 examples, Frank-Wolfe and subgrdient method outperform Gurobi across different problem sizes by showing a consistent saving of 96% and, respectively, 70% of computation time.

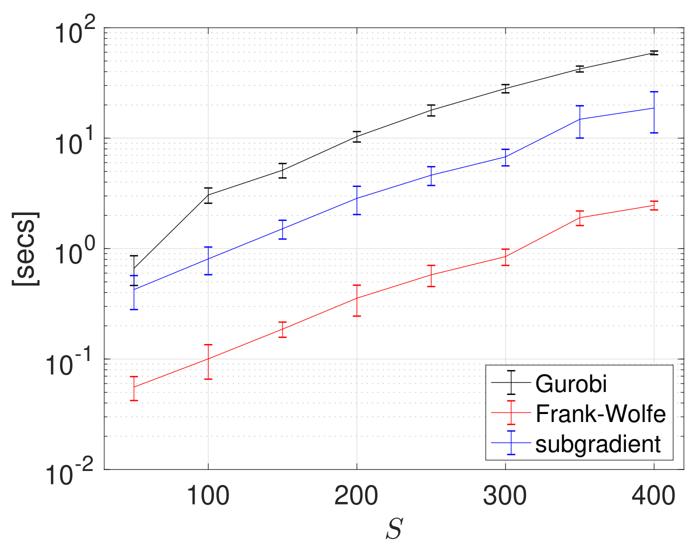


Figure 2.4: Average computation time and 3-standard deviation intervals of 1000 experiments with $T = A = 10$.

2.4 Related work and remarks

The results in this chapter is closely related to two separate research topic: MDP and network equilibrium. We already covered basic MDP model in §2.1.1. In this section, we first briefly review the basic results in network equilibrium problems, then discuss how MDP and network equilibrium was first combined together.

2.4.1 Network equilibrium

The theory of network equilibrium dates back to the famous Wardrop equilibrium principle [177] for vehicles using different routes in a transportation network, which says *the journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.* Over the years, the Wardrop equilibrium principle has become the most important theoretical tool in transportation network design and evaluation [61, 62, 38]; we refer interested readers to Michael Patricksson's book [137], where some key topics in this chapter, *e.g.*. variable demand variation, Frank-Wolfe and projected subgradient methods for network equilibrium are all discussed in the context of classical network equilibrium model; see its Sec.2.2.3, Sec.4.1.3, Sec.4.3.7 for relevant details. Meanwhile, network equilibrium problems also attracted attention in convex optimization community, due to its elegant interpretation of duality theory. In fact, in the preface of his book [17], Tyrrell Rochafellar commented that "monotropic programming problems enjoy a remarkably complete and symmetric duality theory, almost every bit as constructively useful as the one for linear programming".

We briefly review the linear optimal distribution and differential problems in [150, Sec. 7A, Sec. 7C] and Optimal distribution and differential in [150, Sec. 8D, 8G] in the following. We encourage the readers to compare them against the materials presented in § 2.1.1 and § 2.1.2.

2.4.1.1 Linear optimal distribution and differential

Consider a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each edge $e \in \mathcal{E}$ is associated with a cost c_e, a typical linear optimal distribution and, respectively, linear optimal differential problem is defined as following two linear programming

$$\begin{aligned} \underset{y \in \mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} \quad & \langle c, y \rangle \\ \text{subject to} \quad & D(\mathcal{G})y = b, \;\; y \geqslant 0. \end{aligned} \tag{2.38}$$

$$\begin{aligned} \underset{v \in \mathbb{R}^{|\mathcal{V}|}}{\text{maximize}} \quad & -\langle b, v \rangle \\ \text{subject to} \quad & D(\mathcal{G})^\top v + c \geqslant 0, \end{aligned} \tag{2.39}$$

where $b^\top \mathbf{1}_{|\mathcal{V}|} = 0$ and $c \in \mathbb{R}^{|\mathcal{E}|}_{++}$.

Using results in §A.2 one can verify that the KKT conditions for (2.38) and (2.39) are identical, given by

$$D(\mathcal{G})y = b, \;\; y \geqslant 0, \tag{2.40a}$$

$$\mu \geqslant 0, \tag{2.40b}$$

$$c + D(\mathcal{G})^\top v - \mu = 0, \tag{2.40c}$$

$$y_e \mu_e = 0, \;\; \forall e \in \mathcal{E}. \tag{2.40d}$$

Optimization (2.38) and (2.39) are also known as the shortest path problem. To see this, let graph $\mathcal{G}$ be connected, and elements in vector $b \in \mathbb{R}^{|\mathcal{V}}$ are all zero except two: $b_i = 1$ and $b_j = -1$ for a distinct pair of nodes $i, j \in \mathcal{V}$. Then solution to (2.38) and (2.39) gives an shortest path between node i and node j where the length of edge are given by vector $c \in \mathbb{R}^{|\mathcal{E}}$, in the following sense. Let $y^\star \in \mathbb{R}^{|\mathcal{E}|}$ and $v^\star \in \mathbb{R}^{|\mathcal{V}|}$ be an optimal solution to (2.38) and, respectively, (2.39). If $y_e^\star > 0$, then edge $e \in \mathcal{E}$ is on a shortest path (not necessarily unique) from node i to node j. Further, $v_i^\star - v_j^\star$ gives the length of any shortest path from node i to node j.

2.4.1.2 Optimal distribution and differential

The natural extension of optimization (2.38) and (2.39) is the optimal distribution and differential problem, given by the following

$$\begin{aligned} &\underset{y\in\mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} \quad \textstyle\sum_{e\in\mathcal{E}} \displaystyle\int_0^{y_e} \phi_e(\alpha)\alpha \\ &\text{subject to} \quad D(\mathcal{G})y = b, \ \ y \geqslant 0. \end{aligned} \tag{2.41}$$

$$\begin{aligned} &\underset{v\in\mathbb{R}^{|\mathcal{V}|},u\in\mathbb{R}^{|\mathcal{E}|}}{\text{maximize}} \quad -\langle b, v\rangle - \textstyle\sum_{e\in\mathcal{E}} \displaystyle\int_{\phi_e(0)}^{u_e} \phi_e^{-1}(\alpha)d\alpha \\ &\text{subject to} \quad D(\mathcal{G})^\top v + u \geqslant 0, \end{aligned} \tag{2.42}$$

where $b^\top \mathbf{1}_{|\mathcal{V}|} = 0$ and $\phi_e : \mathbb{R}_+\mathbb{R}_+$ is an continuously increasing[4] function. The KKT conditions of (2.41) and (2.39) are also identical and given by

$$D(\mathcal{G})y = b, \ \ y \geqslant 0, \tag{2.43a}$$

$$\mu \geqslant 0, \tag{2.43b}$$

$$\phi(y) + D(\mathcal{G})^\top v - \mu = 0, \tag{2.43c}$$

$$y_e\mu_e = 0, \ \ e \in \mathcal{E}. \tag{2.43d}$$

Conditions in (2.43) are also know as the network equilibrium[5] condition. A particular neat interpretation of (2.43) is via resistor circuits as follows. Consider a resistor circuits network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ denotes a pin with electrical potential v_i, and edge $e \in \mathcal{E}$ denotes a nonlinear resistor with electrical current y_e and the current-voltage relation characterized by continuously increasing function ϕ_e. Then condition in (2.43) are exactly the Kirchoff current and voltage laws, where $y_e = 0$ (or equivalently, $\mu_e > 0$) indicates that the resistor on edge $e \in \mathcal{E}$ is *shorted* at equilibrium.

---

[4]In general, function ϕ_e only needs to be non-decreasing; see [150, Sec. 8B].

[5]See [150, Sec. 8H] for a detailed discussion.

2.4.2 MDP routing games

MDP has been one of the most well-studied stochastic decision-making model; see [141, 16] for examples. In particular,

Inspired by recent advances in mean field games mean field games over a graph [67, 68, 69, 70], Calderone & Sastry [30] first combined the concept of MDP and network equilibrium, and proposed the so-called MDP routing games, whose Wardrop equilibrium is introduced in §2.1.2. However, only the primal problem (2.7) was studied in [30], without the dual problem (2.8), or any of the variable-demand and multi-commodity variations in §2.2. Furthermore, only off-the-shelf optimization software was used in [30]. These shortcomings are addressed later in [193], and their results are covered in §2.1 and §2.2. The results in [30] has been also extended to infinite-horizon averaged cost MDP routing games in [29], and the cases where the equilibrium is subject to side constraints and tolling [106].

Future work As a first step towards general stochastic dynamic network equilibrium problems, the Markovian network equilibrium problems introduces in this chapter combines two separate and otherwise almost independent research area: MDP together with classical network equilibrium. However, such combination is by no means thorough and many possible extensions are still open questions. For example, the connection between partially observable MDP, a natural extension of MDP, to the Markovian network equilibrium problems are still unclear. Another future direction is to apply the Markovian network equilibrium model to incentive design, where the goal is to shift the equilibrium to more desirable point by properly design the supply functions.

3 PI control and network optimization

We start this chapter with a trajectory optimization problem as follows. Imagine you are controlling a three-wheeled rover; see Figure 3.1 for an illustration. Your goal is to design a sequence of inputs on its three motors such that

- the rover's position will follow a reference trajectory while consuming as least amount of battery as possible

- the rover's position, velocity and acceleration will respect a given set of constraints

What mathematical model will help you achieve the above requirements?

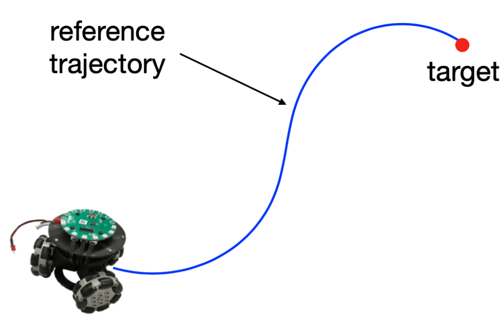


Figure 3.1: Tracking a reference trajectory

The answer is given by the following finite-horizon discrete time constrained optimal control problem

$$\begin{array}{ll} \underset{u_{[0,L-1]},x_{[0,L]}}{\text{minimize}} & \frac{1}{2}\sum_{t=1}^{L}(x_t - r_t)^\top Q_t(x_t - r_t) + \frac{1}{2}\sum_{t=0}^{L-1} u_t^\top R_t u_t \\ \text{subject to} & x_t = Ax_{t-1} + Bu_{t-1}, \;\; x_0 = \hat{x}, \\ & u_t \in \mathbb{U}_t, \; x_{t+1} \in \mathbb{X}_{t+1}, \;\; t \in [0, L-1] \end{array} \tag{3.1}$$

where one aims to find an optimal sequence of inputs $u_0, \dots, u_{L-1} \in \mathbb{R}^m$ and state $x_1, \dots, x_L \in \mathbb{R}^m$ of a linear time invariant system $x_t = Ax_{t-1} + Bu_{t-1}$ with given initial state $x_0 \in \mathbb{R}^n$ that minimizes a quadratic tracking cost $\frac{1}{2}\sum_{t=1}^{L}(x_t - r_t)^\top Q_t(x_t - r_t) + \frac{1}{2}\sum_{t=0}^{L-1} u_t^\top R_t u_t$ with respect to a reference trajectory $\{r_t\}_{t=1}^{L}$ while subject to convex input and state constraint $u_{t-1} \in \mathbb{U}_{t-1}$, $x_t \in \mathbb{X}_t$. Here we assume, for all $1 \leqslant t \leqslant L$, that matrices $Q_t \in \mathbb{R}^{n\times n}$ and $R_t \in \mathbb{R}^{m\times m}$ are symmetric and positive semi-definite, and sets $\mathbb{U}_{t-1} \subset \mathbb{R}^m$ and $\mathbb{X}_t \subset \mathbb{R}^n$ are convex. Efficient solution methods to optimization (3.1) are not only the backbone of model predictive control theory [120, 119, 142, 91, 55], but also an integral part in non-convex trajectory optimization where it serves as a convex subproblem that approximates the original non-convex problem.

How can we design efficient optimization algorithms for problem (3.29)? As a rule of thumb, Yurri Nesterov once wrote that[1] "efficient optimization methods can be developed only by intelligently employing the structure of particular instances of problems". In this chapter, we will design an algorithm using the exact principle. In particular, our algorithm is motivated by the following structure of optimization (3.1)

Without its affine equality constraints, optimization (3.1) *decompose into* $2L$ *separate optimization with lower dimensional variables.*

The reader may already noticed: every optimization problems discussed in Chapter 2 fits a similar description. Indeed, minimizing a separable convex function subject to affine constraints is the characteristic feature of network optimization [150, 17]. In the following, we will start with another class of optimization problem in §3.1 that fits the above description, called distributed optimization, where we design efficient algorithms based on RLC circuits dynamics. We then use similar ideas to design a novel first order primal-dual method for problem (3.1), termed proportional-integral projected gradient method in §3.2. We end this chapter with two brief review of related work in distributed optimiztaion and trajectory optimiztaion literature in §3.3 and, respectively, §3.4.

---

[1]See the preface of [131]

3.1 Distributed optimization

In this section, we consider a special class of network optimization problems, named *distributed optimization*, which aims to minimize the sum of cost functions over a connected graph using local computation and communication only.

3.1.1 RC circuits and distributed consensus

We start with a special–and perhaps the most well studied–case of distributed optimization problems, termed *distributed consensus*, described as follows.

Problem 3.1. *Consider an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ maintains a variable x_i that takes value $x_i^k \in \mathbb{R}^n$ at discrete time $k \in \mathbb{N}$. Design an update rule such that 1) information exchange between node i and j is allowed only if $\{i, j\} \in \mathcal{E}$, and 2) $\lim_{k\to\infty} x_i^k - x_j^k = 0$ for all $\{i, j\} \in \mathcal{E}$.*

To gain some intuitions on Problem 3.1, consider the following toy example. Imagine a group of people are sitting in a noisy room. Each person has an opinion on how many papers a PhD student should read every week. Since the room is noisy, each person can only talk to people sitting next to him/her. Now, assuming everyone is open minded and willing to change his/her opinion based on his/her "neighbors", then eventually the group will reach a consensus on the number of papers a student should read. Then the process of people's opinions reaching consensus can be viewed as solving an instance of Problem 3.1, where nodes in $\mathcal{V}$ denote different people in the room, edges in $\mathcal{E}$ denote the adjacency relation of people's seats, and different x_i denotes different people's opinion.

A simple and very effective solution to the distributed consensus problem is based on the following ordinary differential equation (ODE)

$$\frac{d}{dt}x_i(t) = -\sum_{j\in\mathcal{N}(i)} r(x_i(t) - x_j(t)), \ \ \forall i \in \mathcal{V}, \tag{3.2}$$

where $r \in \mathbb{R}_{++}$, and $j \in \mathcal{N}(i)$ if and only if $\{i, j\} \in \mathcal{E}$. Let

$$E(\mathcal{G}) = D(\mathcal{G}) \otimes I_n, \tag{3.3}$$

then (3.2) can be written compactly as

$$\frac{d}{dt}x(t) = -rE(\mathcal{G})E(\mathcal{G})^\top x(t) \tag{3.4}$$

where $x = [x_1^\top, \ldots, x_{|\mathcal{V}|}^\top]^\top$.

An intuitive interpretation of equation (3.4) is the Kirchhoff current law of a *RC circuits* defined as follows. Let each node $i \in \mathcal{V}$ denote a pin with electrical potential $x_i(t)$ at time t. Suppose a) between each pin i and ground (zero potential point), we add a linear capacitor, mapping voltage $x_i(t)$ to current $\frac{d}{dt}x_i(t)$; b) on each edge $\{ij\} \in \mathcal{E}$, we add a linear resistor, mapping voltage $x_i - x_j$ to current $r(x_i - x_j)$; see Fig. 3.2 and Tab. 3.1 for an illustration.

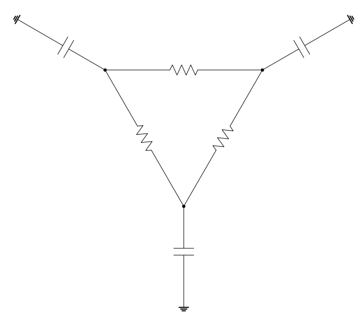


Figure 3.2: An illustration of RC circuits.

device	voltage	current
$\dashv\vdash$	x_i	$\frac{d}{dt}x_i$
$\sim\!\!\sim$	$x_i - x_j$	$r(x_i - x_j)$

Table 3.1: Voltage-current relation of RC units

Finally, in order to obtain a discrete time update rule, a simple approach is to apply

Euler-forward method to (3.4), which gives

$$x^{k+1} = (I - \alpha r E(\mathcal{G}) E(\mathcal{G})^\top) x^k \tag{3.5}$$

where $x^k = x(k\alpha)$ for some $\alpha > 0$. One can verify that if the second largest eigenvalue of matrix $(I - \alpha r E(\mathcal{G}) E(\mathcal{G})^\top)$ lie in the open interval $(-1, 1)$, then (3.5) ensures $x_i(k)$ converges to the nullspace of $E(\mathcal{G}) E(\mathcal{G})^\top$ where $x_i = x_j$ for all $\{i, j\} \in \mathcal{E}$.

3.1.2 RLC circuits and distributed optimization

We are now ready to introduce a general distributed optimization problem, which is described as follows.

Problem 3.2. *Consider an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in \mathcal{V}$ maintains a variable x_i that takes value $x_i^k \in \mathbb{R}^n$ at discrete time $k \in \mathbb{N}$, a cost function $f_i : \mathbb{R}^n \to \mathbb{R}$, and a common constraint set $\mathbb{X}_0$. Design an update rule such that 1) information exchange between node i and j is allowed only if $\{i, j\} \in \mathcal{E}$, information on f_i is only available to node i for all $i \in \mathcal{V}$, and 2) $\lim_{k\to\infty} x_i^k - x_j^k$ for all $\{i, j\} \in \mathcal{E}$, $\lim_{k\to\infty} x_i^k \in \operatorname{argmin}_{y\in\mathbb{X}_0} \sum_{i\in\mathcal{V}} f_i(y)$ for all $i \in \mathcal{V}$.*

To gain some intuitions on Problem 3.2, consider the following toy example. Imagine a group of blind people are

We write the above distributed optimization problem compactly as follows

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x) := \sum_{i\in\mathcal{V}} f_i(x_i) \\ \text{subject to} & \sqrt{l} E(\mathcal{G})^\top x = 0, \;\; x \in \mathbb{X} := \mathbb{X}_0^{|\mathcal{V}|}, \end{array} \tag{3.6}$$

where $x = [x_1^\top, \ldots, x_{|\mathcal{V}|}^\top]^\top$; $\mathbb{X}_0^{|\mathcal{V}|}$ is the Cartesian product of $|\mathcal{V}|$ copies of closed convex set $\mathbb{X}_0 \subseteq \mathbb{R}^n$; $f_i : \mathbb{X}_0 \to \mathbb{R}$ is a convex differentiable function available to node i only, scalar $l \in \mathbb{R}$

is strictly positive and matrix $E(\mathcal{G})$ is given by

$$E(\mathcal{G}) = D(\mathcal{G}) \otimes I_n. \tag{3.7}$$

Compared with usual optimization, problem (3.6) has the additional implicit requirement that information exchange must obey the connection structure in graph $\mathcal{G}$, and information on each cost function must be queried locally.

We now state our assumptions on problem (3.6) as follows.

Assumption 3.1. *1. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is undirected and connected. Edge weights $l, r \in \mathbb{R}_{++}$ are strictly positive, and let $\sigma \in \mathbb{R}_{++}$ such that $E(\mathcal{G})E(\mathcal{G})^\top \preceq \sigma I$*

2. $\mathbb{X}_0 \subseteq \mathbb{R}^n$ is a closed convex set,

3. For all $i \in \mathcal{V}$, $f_i : \mathbb{X}_0 \to \mathbb{R}$ is continuously differentiable, convex and λ-smooth, i.e., both f_i and $\frac{\lambda}{2}\|\cdot\|_2^2 - f_i$ are convex over $\mathbb{X}_0$. There exists $x^\star, u^\star$ such that

$$\sqrt{l}E(\mathcal{G})^\top x^\star = 0, \;\; x^\star \in \mathbb{X} = \mathbb{X}_0^{|\mathcal{V}|}, \tag{3.8a}$$

$$\langle \sqrt{l}E(\mathcal{G})u^\star + \nabla f(x^\star), x - x^\star \rangle \geqslant 0, \;\; \forall x \in \mathbb{X}. \tag{3.8b}$$

Inspired by the role of ODE (3.4) in distributed consensus problem, we consider the following ODE for distributed optimization

$$\begin{aligned} \frac{d}{dt}\nabla\psi(x(t)) =& - rE(\mathcal{G})E(\mathcal{G})^\top x(t) - \sqrt{l}E(\mathcal{G})u(t) - \nabla f(x(t)), \\ \frac{d}{dt}u(t) =& \sqrt{l}E(\mathcal{G})^\top x(t), \end{aligned} \tag{3.9}$$

where

$$x = \begin{bmatrix} x_1^\top \\ \vdots \\ x_{|\mathcal{V}|}^\top \end{bmatrix}, \;\; \nabla f(x) = \begin{bmatrix} \nabla f_1(x_1) \\ \vdots \\ \nabla f_{|\mathcal{V}|}(x_{|\mathcal{V}|}) \end{bmatrix}, \;\; \nabla\psi(x) = \begin{bmatrix} \nabla\psi_0(x_1) \\ \vdots \\ \nabla\psi_0(x_{|\mathcal{V}|}) \end{bmatrix}$$

and $\psi_0 : \mathbb{X}_0 \to \mathbb{R}$ is a smooth function to be specified later. Similar to (3.4), the above ODE can also be interpreted as the Kirchhoff current law of a *RLC circuits* as follows. Let each node $i \in \mathcal{V}$ denote a pin with electrical potential $x_i(t)$ at time t. Suppose a) between each pin i and ground (zero potential point), we add a nonlinear capacitor in parallel with a nonlinear resistor, mapping voltage $x_i(t)$ to current $\frac{d}{dt}\nabla\psi_0(x_i(t))$ for a differentiable function ψ_0 and, respectively, $\nabla f_i(x_i)$; b) on each edge $\{ij\} \in \mathcal{E}$, we add a linear inductor in parallel with a linear resistor, mapping voltage $x_i - x_j$ to current $l\int_0^t (x_i(\tau) - x_j(\tau))d\tau$ and, respectively, $r(x_i - x_j)$; see Fig. 3.3 and Tab. 3.2 for an illustration.

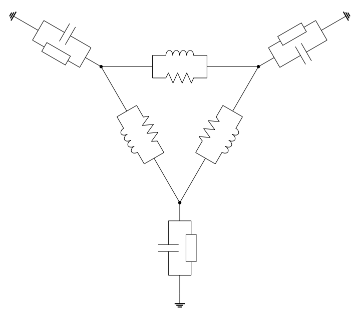


Figure 3.3: An illustration of RLC circuits.

type	symbol	voltage	current
resistor	─▭─	x_i	$\nabla f_i(x_i)$
capacitor	─┤├─	x_i	$\frac{d}{dt}\nabla\psi_0(x_i)$
resistor	─\/\/\/─	$x_i - x_j$	$r(x_i - x_j)$
inductor	─⌒⌒⌒─	$x_i - x_j$	$\int l(x_i - x_j)dt$

Table 3.2: Voltage-current relation of RLC units

Unless function f and ψ are both quadratic function (which will be the focus for the next section), ODE (3.9) will always contain nonlinear terms, which pose a unique challenge in discretization. In particular, applying Euler-forward discretization to (3.9) with step size α

gives the following

$$\nabla\psi(x^{k+1}) = \nabla\psi(x^k) - \alpha(rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k)), \tag{3.10a}$$

$$u^{k+1} = u^k + \alpha\sqrt{l}E(\mathcal{G})^\top x^{k+1} \tag{3.10b}$$

where, in order to simplify our notation, we denote $x(k\alpha)$ as x^k, $u(k\alpha)$ as u^k for all $k \in \mathbb{N}$. Notice that we use x^{k+1} instead of x^k in (3.10b). This technique is known as Gauss-Seidel pass, which is commonly used in similar primal-dual algorithms.

We now start to see the effect of nonlinearity on discretization: the x-update in (3.10a) is a nonlinear equation, and to obtain the value of x^{k+1} we need to apply the "inverse" of $\nabla\psi$ to the right hand side of (3.10a). And the difficulty of computing such inverse depends heavily on the assumption on function ψ. Here we consider a special class of function described as follows.

Assumption 3.2. *Suppose function $\psi : \mathbb{X} \to \mathbb{R}$ is both closed, proper, convex, continuously differentiable over $\mathbb{X}$. Further, $\psi - \frac{1}{2}\|\cdot\|_2^2$ is also convex over $\mathbb{X}$.*

With the above assumption, we can compute the "inverse" of $\nabla\psi$ using *Fenchel-Young theorem*. Here we state a simplified version of the theorem in the following lemma; see [152, Thm. 23.5] for full details[2].

In addition, since ψ is 1-strongly convex, it is straightforward to show that $\min_{y\in\mathbb{X}} -\langle y, \nabla\psi(x)\rangle + \psi(y)$ has a unique minimizer, and combining (3.10a) with the above lemma we can show the following

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_{x\in\mathbb{X}} \alpha\langle w^k, x\rangle + \psi(x) - \langle\nabla\psi(x^k), x\rangle \\ &= \operatorname*{argmin}_{x\in\mathbb{X}} \alpha\langle w^k, x\rangle + \psi(x) - \psi(x^k) - \langle\nabla\psi(x^k), x - x^k\rangle \\ &= \operatorname*{argmin}_{x\in\mathbb{X}} \alpha\langle w^k, x\rangle + B_\psi(x, x^k) \end{aligned} \tag{3.11}$$

[2]In the author's humble opinion, [152, Thm. 23.5] is one of the most elegant and profound results in convex analysis.

where $w^k = rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k)$. Notice that the second step is because $-\psi(x^k) + \langle \nabla\psi(x^k), x^k\rangle$ is independent of x, hence does not change the argmin. The last step uses the definition of Bregman divergence in (B.4).

By replacing (3.10a) with (3.11), we arrive at the following algorithm.

$$
\begin{aligned}
x^{k+1} =& \operatorname*{argmin}_{x\in\mathbb{X}}\ \alpha\langle rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k), x\rangle + B_\psi(x, x^k), && (3.12a)\\
u^{k+1} =& u^k + \alpha\sqrt{l}E(\mathcal{G})^\top x^{k+1}. && (3.12b)
\end{aligned}
$$

The following theorem shows that the above algorithm indeed solves problem (3.6). We will use the following inequalities implied by Assumption 3.1 and Assumption 3.2; see Appendix B for relevant details

$$
\begin{aligned}
0 \leqslant& B_f(x', x) = f(x') - f(x) - \langle \nabla f(x), x' - x\rangle \leqslant \frac{\lambda}{2}\left\|x' - x\right\|_2^2 && (3.13a)\\
\frac{1}{2}\left\|x' - x\right\|_2^2 \leqslant& B_\psi(x', x) = \psi(x') - \psi(x) - \langle \nabla\psi(x), x' - x\rangle && (3.13b)
\end{aligned}
$$

Theorem 3.1. *Suppose Assumption 3.1 and Assumption 3.2 hold. Let sequence* $\{x^k, u^k\}$ *be generated by* (3.12) *where* $0 < \alpha \leqslant \min\{\frac{1}{\lambda + r\sigma}, \frac{r}{l}\}$, *then*

$$
\begin{aligned}
\frac{r}{2}\left\|E(\mathcal{G})^\top \hat{x}^k\right\|_2^2 \leqslant& \frac{B_\psi(x^\star, x^1) + \frac{1}{2}\left\|u^1 - u^\star\right\|_2^2}{\alpha k},\\
B_f(\tilde{x}^k, x^\star) \leqslant& \frac{B_\psi(x^\star, x^1) + \frac{1}{2}\left\|u^1 - u^\star\right\|_2^2}{\alpha k}
\end{aligned}
$$

where $\hat{x}^k = \frac{1}{k}\sum_{j=1}^k x^j$, $\tilde{x}^k = \frac{1}{k}\sum_{j=1}^k x^{j+1}$.

Proof. Step 1, applying Theorem A.2 to (3.12a) gives

$$\begin{aligned}
0 \leqslant & \langle rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k) + \frac{1}{\alpha}(\nabla\psi(x^{k+1}) - \nabla\psi(x^k)), x^\star - x^{k+1}\rangle \\
\overset{(3.8a)}{=} & r\langle E(\mathcal{G})^\top(x^k - x^\star), E(\mathcal{G})^\top(x^\star - x^{k+1})\rangle + \sqrt{l}\langle u^k, E(\mathcal{G})^\top(x^\star - x^{k+1})\rangle + \langle \nabla f(x^k), x^\star - x^{k+1}\rangle \\
& + \frac{1}{\alpha}\langle \nabla\psi(x^{k+1}) - \nabla\psi(x^k), x^\star - x^{k+1}\rangle
\end{aligned} \tag{3.14}$$

Similarly, (3.8b) directly implies that

$$0 \leqslant \langle \sqrt{l}E(\mathcal{G})u^\star + \nabla f(x^\star), x^{k+1} - x^\star\rangle \tag{3.15}$$

Step 2, one can verify the following three inequalities by applying the "three point property" in (B.5) to function $f(x)$, $\psi(x)$, $\frac{1}{2}\left\|E(\mathcal{G})^\top x\right\|_2^2$ and, respectively, $\frac{1}{2}\left\|u\right\|_2^2$,

$$\langle \nabla f(x^\star) - \nabla f(x^k), x^{k+1} - x^\star\rangle = B_f(x^{k+1}, x^k) - B_f(x^{k+1}, x^\star) - B_f(x^\star, x^k), \tag{3.16a}$$

$$\langle \nabla\psi(x^{k+1}) - \nabla\psi(x^k), x^\star - x^{k+1}\rangle = B_\psi(x^\star, x^k) - B_\psi(x^\star, x^{k+1}) - B_\psi(x^{k+1}, x^k), \tag{3.16b}$$

$$\frac{1}{2}\left\|u^{k+1} - u^\star\right\|_2^2 - \frac{1}{2}\left\|u^{k+1} - u^k\right\|_2^2 - \frac{1}{2}\left\|u^k - u^\star\right\|_2^2 = \langle u^k - u^\star, u^{k+1} - u^k\rangle \tag{3.16c}$$

$$\begin{aligned}
& \langle E(\mathcal{G})^\top(x^\star - x^k), E(\mathcal{G})^\top(x^{k+1} - x^\star)\rangle \\
= & \frac{1}{2}\left\|E(\mathcal{G})^\top(x^{k+1} - x^k)\right\|_2^2 - \frac{1}{2}\left\|E(\mathcal{G})^\top x^{k+1}\right\|_2^2 - \frac{1}{2}\left\|E(\mathcal{G})^\top x^k\right\|_2^2,
\end{aligned} \tag{3.16d}$$

where we also used (3.8a) in (3.16d). Combining (3.12b), (3.16c) and (3.8a) we obtain

$$\frac{1}{2}\left\|u^{k+1} - u^\star\right\|_2^2 - \frac{1}{2}\left\|u^k - u^\star\right\|_2^2 - \frac{(\alpha)^2 l}{2}\left\|E(\mathcal{G})^\top x^{k+1}\right\|_2^2 = \alpha\sqrt{l}\langle u^k - u^\star, E(\mathcal{G})^\top(x^{k+1} - x^\star)\rangle \tag{3.17}$$

Step 3, based on Assumption 3.1 and Assumption 3.2, the Bregman divergence of function f, ψ and $\frac{1}{2}\left\|E(\mathcal{G})^\top x\right\|$ admit the following bounds

$$0 \leqslant B_f(x^\star, x^k) \tag{3.18a}$$

$$B_f(x^{k+1}, x^k) \leqslant \frac{\lambda}{2} \left\|x^{k+1} - x^k\right\|_2^2 \tag{3.18b}$$

$$\frac{1}{2} \left\|x^{k+1} - x^k\right\|_2^2 \leqslant B_\psi(x^{k+1}, x^k) \tag{3.18c}$$

$$\frac{r}{2} \left\|E(\mathcal{G})^\top (x^{k+1} - x^k)\right\|_2^2 \leqslant \frac{r\sigma}{2} \left\|x^{k+1} - x^k\right\|_2^2 \tag{3.18d}$$

Summing up (3.14), (3.15), (3.16a), $\frac{1}{\alpha}$(3.16b), $r\times$(3.16d), $\frac{1}{\alpha}\times$(3.17), (3.18a), (3.18b), $\frac{1}{\alpha}\times$(3.18c) and (3.18d), then using the assumption that $0 < \alpha \leqslant \min\{\frac{1}{\lambda + r\sigma}, \frac{r}{l}\}$ and letting $k = j$ we obtain the following

$$\frac{r}{2} \left\|E(\mathcal{G})^\top x^j\right\|_2^2 + B_f(x^{j+1}, x^\star) \leqslant \frac{1}{\alpha}(V^j - V^{j+1})$$

for all $j \geqslant 1$, where $V^j = B_\psi(x^\star, x^j) + \frac{1}{2}\left\|u^j - u^\star\right\|_2^2$. Summing up the above equation from $j = 1$ to $j = k$ we have

$$\sum_{j=1}^k \frac{r}{2} \left\|E(\mathcal{G})^\top x^j\right\|_2^2 + \sum_{j=1}^k B_f(x^{j+1}, x^\star) \leqslant \frac{1}{\alpha}(V^1 - V^2 + \cdots + V^k - V^{k+1}) = \frac{1}{\alpha}(V^1 - V^{k+1}) \leqslant \frac{1}{\alpha} V^1.$$

Since $\frac{r}{2}\left\|E(\mathcal{G})^\top x\right\|_2^2$ and $B_f(x, x^\star)$ are non-negative functions, the above equation implies that

$$\sum_{j=1}^k \frac{r}{2} \left\|E(\mathcal{G})^\top x^j\right\|_2^2 \leqslant \frac{1}{\alpha} V^1, \quad \sum_{j=1}^k B_f(x^{j+1}, x^\star) \leqslant \frac{1}{\alpha} V^1.$$

Step 4, applying Jensen's inequality (A.4) to the two inequalities above completes the proof. □

In the following, we introduce two variations to algorithm 3.12 based on two popular generalizations to gradient descent method: stochastic gradient descent and proximal gradient method.

3.1.2.1 Noisy gradients

We now consider the case where the gradient $\nabla f(x^k)$ used in (3.12) are corrupted by an independent noise vector $\eta^k \in \mathbb{R}^n$ with zero mean and bounded variance, *i.e.*, $\mathbb{E}[\eta^k] = 0$ and $\mathbb{E}[\|\eta\|_2^2] \leqslant \gamma^2$ for some $\gamma > 0$. We will see that this additive noise will slow down the convergence of algorithm (3.12).

$$
\begin{aligned}
x^{k+1} =& \operatorname*{argmin}_{x\in\mathbb{X}} \alpha^k \langle rL(\mathcal{G})x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k) + \eta^k, x\rangle + B_\psi(x, x^k), && (3.19a)\\
u^{k+1} =& u^k + \alpha^k \sqrt{l} E(\mathcal{G})^\top x^{k+1}. && (3.19b)
\end{aligned}
$$

where α^k is a step size varies with iteration counter k. If $\alpha^k \equiv \alpha$ and $\eta^k \equiv 0$ for all k, then (3.19) reduces to (3.12).

We now prove the key property of iterations in (3.19).

Theorem 3.2. *Suppose Assumption 3.1 and* (3.2) *hold. Let sequence* $\{x^k, u^k\}$ *be generated by* (3.19) *where* $0 < \alpha^k \leqslant \min\{\frac{1}{2(\lambda+r\sigma)}, \frac{r}{l}\}$ *and* η^k *is an independent random variable with* $\mathbb{E}[\eta^k] = 0$ *and* $\mathbb{E}[\|\eta\|_2^2] \leqslant \gamma^2$ *for all* $k \geqslant 1$*, then*

$$
\begin{aligned}
\frac{r}{2}\mathbb{E}[\left\|E(\mathcal{G})^\top \hat{x}^k\right\|_2^2] \leqslant& \frac{B_\psi(x^\star, x^1) + \frac{1}{2}\left\|u^1 - u^\star\right\|_2^2 + \gamma^2 \sum_{j=1}^k (\alpha^j)^2}{\sum_{j=1}^k \alpha^j},\\
\mathbb{E}[B_f(\tilde{x}^k, x^\star)] \leqslant& \frac{B_\psi(x^\star, x^1) + \frac{1}{2}\left\|u^1 - u^\star\right\|_2^2 + \gamma^2 \sum_{j=1}^k (\alpha^j)^2}{\sum_{j=1}^k \alpha^j}
\end{aligned}
$$

where $\hat{x}^k = \frac{1}{\sum_{j=1}^k \alpha^j}\sum_{j=1}^k \alpha^j x^j$, $\tilde{x}^k = \frac{1}{\sum_{j=1}^k \alpha^j}\sum_{j=1}^k \alpha^j x^{j+1}$

Proof. First, applying Theorem A.2 to (3.19a) gives

$$
\begin{aligned}
0 \leqslant& \langle rL(\mathcal{G})x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k) + \eta^k + \frac{1}{\alpha^k}(\nabla\psi(x^{k+1}) - \nabla\psi(x^k)), x^\star - x^{k+1}\rangle\\
\overset{(3.8a)}{=}& r\langle L(\mathcal{G})(x^k - x^\star), x^\star - x^{k+1}\rangle + \sqrt{l}\langle u^k, E(\mathcal{G})^\top(x^\star - x^{k+1})\rangle + \langle \nabla f(x^k), x^\star - x^{k+1}\rangle \quad (3.20)\\
&+ \frac{1}{\alpha^k}\langle \nabla\psi(x^{k+1}) - \nabla\psi(x^k), x^\star - x^{k+1}\rangle + \langle \eta^k, x^\star - x^{k+1}\rangle
\end{aligned}
$$

Notice that equalities in (3.16) always hold. Combining (3.19b), (3.16c) and (3.8a) we obtain

$$\frac{1}{2}\left\|u^{k+1}-u^\star\right\|_2^2-\frac{1}{2}\left\|u^k-u^\star\right\|_2^2-\frac{(\alpha^k)^2 l}{2}\left\|E(\mathcal{G})^\top x^{k+1}\right\|_2^2=\alpha^k\sqrt{l}\langle u^k-u^\star,E(\mathcal{G})^\top(x^{k+1}-x^\star)\rangle \tag{3.21}$$

In addition, inequalities in (3.18) also hold under Assumption 3.1 and Assumption 3.2.

Next, summing up (3.20), (3.15), (3.16a), $\frac{1}{\alpha^k}$(3.16b), $r\times$(3.16d), $\frac{1}{\alpha^k}\times$(3.21), (3.18a), (3.18b), $\frac{1}{\alpha^k}\times$(3.18c) and (3.18d), then using the assumption that $\alpha=\frac{r}{l}$ we obtain the following

$$\begin{aligned}&\frac{r}{2}\left\|E(\mathcal{G})^\top x^k\right\|_2^2+B_f(x^{k+1},x^\star)\\ \leqslant&\frac{1}{\alpha^k}(B_\psi(x^\star,x^k)+\frac{1}{2}\left\|u^k-u^\star\right\|_2^2)-\frac{1}{\alpha^k}(B_\psi(x^\star,x^{k+1})+\frac{1}{2}\left\|u^{k+1}-u^\star\right\|_2^2)\\ &+(\frac{\lambda+r\sigma}{2}-\frac{1}{2\alpha^k})\left\|x^{k+1}-x^k\right\|_2^2+\langle\eta^k,x^\star-x^{k+1}\rangle\end{aligned} \tag{3.22}$$

Notice that

$$\begin{aligned}\langle\eta^j,x^\star-x^{j+1}\rangle=&-\langle\eta^j,x^{j+1}-x^j\rangle-\langle\eta^j,x^j-x^\star\rangle\\ =&\frac{\varepsilon^j}{2}\left\|\eta^j\right\|_2^2+\frac{1}{2\varepsilon^j}\left\|x^{j+1}-x^j\right\|_2^2-\langle\eta^j,x^j-x^\star\rangle\end{aligned}$$

where $\varepsilon^j=\frac{\alpha^j}{1-\alpha^j(\lambda+r\sigma)}>0$ as $\alpha^j\leqslant\frac{1}{2(\lambda+r\sigma)}$. Summing up the above inequality together with (3.22) and letting $k=j$ we have

$$\begin{aligned}\frac{r}{2}\left\|E(\mathcal{G})^\top x^j\right\|_2^2+B_f(x^{j+1},x^\star)\leqslant&\frac{1}{\alpha^j}V^j-\frac{1}{\alpha^j}V^{j+1}+\frac{\epsilon^j}{2}\left\|\eta^j\right\|_2^2-\langle\eta^j,x^j-x^\star\rangle\\ \leqslant&\frac{1}{\alpha^j}V^j-\frac{1}{\alpha^j}V^{j+1}+\alpha^j\left\|\eta^j\right\|_2^2-\langle\eta^j,x^j-x^\star\rangle\end{aligned}$$

where $V^j=B_\psi(x^\star,x^j)+\frac{1}{2}\left\|u^j-u^\star\right\|_2^2$ and we used the fact that $\varepsilon^j=\frac{\alpha^j}{1-\alpha^j(\lambda+r\sigma)}\leqslant 2\alpha^j$ when $\alpha^j\leqslant\frac{1}{2(\lambda+r\sigma)}$. Multiple the above inequality with α^j then taking expectation on both sides we obtain

$$\begin{aligned}&\frac{r\alpha^j}{2}\mathbb{E}\big[\,\big\|E(\mathcal{G})^\top x^j\big\|_2^2\,\big] + \alpha^j\mathbb{E}[B_f(x^{j+1},x^\star)]\\ \leqslant&\mathbb{E}[V^j] - \mathbb{E}[V^{j+1}] + (\alpha^j)^2\mathbb{E}[\big\|\eta^j\big\|_2^2] - \alpha^j\mathbb{E}[\langle\eta^j, x^j - x^\star\rangle]\end{aligned}$$

Since η^k is an independent random variable with $\mathbb{E}[\eta^k] = 0$, we know $\mathbb{E}[\langle\eta^j, x^j - x^\star\rangle] = 0$. Substituting this and $\mathbb{E}[\|\eta\|_2^2] \leqslant \gamma^2$ into the above inequality we have

$$\frac{r\alpha^j}{2}\mathbb{E}\big[\,\big\|E(\mathcal{G})^\top x^j\big\|_2^2\,\big] + \alpha^j\mathbb{E}[B_f(x^{j+1},x^\star)] \leqslant \mathbb{E}[V^j] - \mathbb{E}[V^{j+1}] + (\alpha^j)^2\gamma^2$$

Summing up the above inequality from $j = 1$ to $j = k$ gives

$$\sum_{j=1}^{k}\frac{r\alpha^j}{2}\mathbb{E}\big[\,\big\|E(\mathcal{G})^\top x^j\big\|_2^2\,\big]+\sum_{j=1}^{k}\alpha^j\mathbb{E}[B_f(x^{j+1},x^\star)] \leqslant \mathbb{E}[V^1]-\mathbb{E}[V^{k+1}]+(\alpha^j)^2\gamma^2 \leqslant \mathbb{E}[V^1]+(\alpha^j)^2\gamma^2$$

where the last step is because $\mathbb{E}[V^{k+1}] \geqslant 0$. Since $\mathbb{E}\big[\,\big\|E(\mathcal{G})^\top x^j\big\|_2^2\,\big]$ and $\mathbb{E}[B_f(x^{j+1},x^\star)]$ are both non-negative, the above inequality implies that

$$\begin{aligned}\sum_{j=1}^{k}\frac{r\alpha^j}{2}\mathbb{E}\big[\,\big\|E(\mathcal{G})^\top x^j\big\|_2^2\,\big] \leqslant&\mathbb{E}[V^1] + (\alpha^j)^2\gamma^2\\ \sum_{j=1}^{k}\alpha^j\mathbb{E}[B_f(x^{j+1},x^\star)] \leqslant&\mathbb{E}[V^1] + (\alpha^j)^2\gamma^2\end{aligned}$$

Finally, applying Jensen's inequality (A.4) to the two inequalities above completes the proof.

□

If we choose constant step size $\alpha^k \equiv \alpha$ in (3.19), then Theorem 3.2 converges to a "noise floor" given by $\alpha\gamma^2$, *e.g.*, $\lim_{k\to\infty}\mathbb{E}[B_f(\tilde{x}^k,x^\star)] = \alpha\sigma^2$. If the maximum iteration number k is fixed in advance as a computation budget and we choose $\alpha^k \equiv 1/\sqrt{k}$, then this noise floor also decreases with K at the rate of $\mathcal{O}(1/\sqrt{k})$.

3.1.2.2 Composite objective function

An important paradigm in machine learning is composite objective optimization where the objective function include, in addition to a smooth cost function, a potentially non-smooth regularization function that typically promotes desired solution properties (*e.g.*, sparsity). Examples of composite optimization include ridge regression, lasso, and logistic regression [51]. Combining such paradigm with problem (3.6) gives the following *distributed composite objective optimization problem*

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) + g(x) := \textstyle\sum_{i \in \mathcal{V}} f_i(x_i) + \sum_{i \in \mathcal{V}} g_i(x_i) \\ & E(\mathcal{G})^\top x = 0, \;\; x \in \mathbb{X} := \mathbb{X}_0^{|\mathcal{V}|}. \end{aligned} \tag{3.23}$$

where, in addition to the smooth cost function f_i, each node $i \in \mathcal{V}$ also has a possibly non-smooth convex regularization function $g_i : \mathbb{X}_0 \to \mathbb{R}$.

Motivated by composite objective mirror descent method [51], we propose the following variation of (3.12), which adds an additional regularization to the x-update,

$$x^{k+1} = \underset{x \in \mathbb{X}}{\text{argmin}}\; \alpha \langle r E(\mathcal{G}) E(\mathcal{G})^\top x^k + \sqrt{l} E(\mathcal{G}) u^k + \nabla f(x^k), x \rangle + \alpha g(x) + B_\psi(x, x^k), \tag{3.24a}$$

$$u^{k+1} = u^k + \alpha E_l(\mathcal{G})^\top x^{k+1}. \tag{3.24b}$$

Algorithm (3.24) can be viewed as a distributed extension to composite objective mirror descent method [51]. An important special case of (3.24) is when $\mathbb{X}_0 = \mathbb{R}^n$ and $g_i = \theta \|\cdot\|_1$ with $\theta > 0$ for all $i \in \mathcal{V}$, which aims to induce sparse solution using ℓ_1 norm regularization. In this case, one can show that the x-update in (3.24) reduces to $x^{k+1} = S_{\alpha\theta}(x^k - \alpha w^k)$, where $S_\beta(x)$ is the shrinkage operator defined element-wise as follows [51]

$$[S_\beta(x)]_j = \mathbf{sign}([x]_j) \max\{0, |[x]_j| - \beta\},$$

where $[x]_j$ denote the j-th element of vector x.

There exists $v^\star \in \partial g(x^\star)$ such that

$$\langle \sqrt{l} E(\mathcal{G}) u^\star + \nabla f(x^\star) + v^\star, x - x^\star \rangle \geqslant 0, \ \ \forall x \in \mathbb{X}, \tag{3.25}$$

The following theorem shows that the convergence proof of (3.24) follows the same ideas used by Theorem 3.1.

Theorem 3.3. *Suppose $g_i : \mathbb{X}_0 \to \mathbb{R}^n$ is closed, convex, proper for all $i \in \mathcal{V}$. Suppose Assumption 3.1 holds with* (3.8b) *replaced by* (3.25)*, and Assumption 3.2 holds. Let sequence $\{x^k, u^k\}$ be generated by* (3.24) *where $0 < \alpha \leqslant \min\{\frac{1}{\lambda + r\sigma}, \frac{r}{l}\}$ for all $k \geqslant 1$, then*

$$\begin{aligned} \frac{r}{2} \left\| E(\mathcal{G})^\top \hat{x}^k \right\|_2^2 \leqslant & \frac{B_\psi(x^\star, x^1) + \frac{1}{2} \left\| u^1 - u^\star \right\|_2^2}{\alpha k}, \\ B_f(\tilde{x}^k, x^\star) + B_g(\tilde{x}^k, x^\star) \leqslant & \frac{B_\psi(x^\star, x^1) + \frac{1}{2} \left\| u^1 - u^\star \right\|_2^2}{\alpha k} \end{aligned}$$

where $\hat{x}^k = \frac{1}{k} \sum_{j=1}^k x^j$, $\tilde{x}^k = \frac{1}{k} \sum_{j=1}^k x^{j+1}$.

Proof. Using Theorem A.2 we can show that the x-update in (3.12) implies: there exists $v \in \partial g(x^{k+1})$ such that

$$\begin{aligned} 0 \leqslant & \langle r E(\mathcal{G}) E(\mathcal{G})^\top x^k + \sqrt{l} E(\mathcal{G}) u^k + \nabla f(x^k) + v + \frac{1}{\alpha} (\nabla \psi(x^{k+1}) - \nabla \psi(x^k)), x^\star - x^{k+1} \rangle \\ \overset{(3.8a)}{=} & r \langle E(\mathcal{G}) E(\mathcal{G})^\top (x^k - x^\star), x^\star - x^{k+1} \rangle + \sqrt{l} \langle u^k, E(\mathcal{G})^\top (x^\star - x^{k+1}) \rangle + \langle \nabla f(x^k), x^\star - x^{k+1} \rangle \\ & + \langle v, x^\star - x^{k+1} \rangle + \frac{1}{\alpha} \langle \nabla \psi(x^{k+1}) - \nabla \psi(x^k), x^\star - x^{k+1} \rangle \end{aligned} \tag{3.26}$$

Since $v \in \partial g(x^{k+1})$, we can use (A.5) to show

$$\langle v, x^\star - x^{k+1} \rangle \leqslant g(x^\star) - g(x^{k+1}). \tag{3.27}$$

Further, (3.25) directly implies

$$0 \leqslant \langle \sqrt{l} E(\mathcal{G}) u^\star + \nabla f(x^\star) + v^\star, x^{k+1} - x^\star \rangle \tag{3.28}$$

Again notice that equalities in (3.16) and (3.17) together with inequalities in (3.18) still hold under the current assumptions. Summing up (3.26), (3.27), (3.28), (3.16a), $\frac{1}{\alpha}$(3.16b), $r\times$(3.16d), $\frac{1}{\alpha}\times$(3.17), (3.18a), (3.18b), $\frac{1}{\alpha}\times$(3.18c) and (3.18d), then using the assumption that $\alpha \leqslant \frac{r}{l}$ we obtain the following

$$\frac{r}{2}\left\|E(\mathcal{G})^\top x^j\right\|_2^2 + B_f(x^{j+1}, x^\star) + B_g(x^{j+1}, x^\star) \leqslant \frac{1}{\alpha}(V^j - V^{j+1})$$

for all $j \geqslant 1$, where $V^j = B_\psi(x^\star, x^j) + \frac{1}{2}\left\|u^j - u^\star\right\|_2^2$. Following similar steps as in the proof of Theorem 3.1, we obtain the desired results. □

We finish this section by numerical comparing our algorithms against the distributed mirror descent method in [104, 176, 46] and mirror-prox method (and its composite objective extensions) [128, 86, 76] via numerical examples. We first generate a random graph $\mathcal{G}$ with $|\mathcal{V}| = 30$ and each pair of nodes are connected with probability 0.3. We let $f_i(x_i) = \frac{1}{2}\left\|A_i x_i - b_i\right\|_2^2$ for all $i \in \mathcal{V}$, where entries of A_i and b_i are randomly generated. Using such choice on $\mathcal{G}$ and f, we construct the following two examples (where $n = 30$).

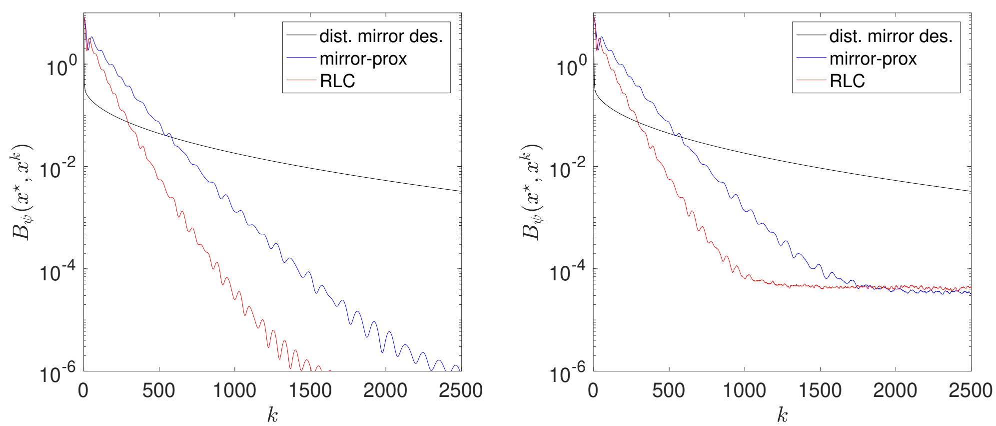
- least squares over simplex: (3.6) with $\mathbb{X}_0 = \{v \in \mathbb{R}^n | v \geqslant 0, \langle v, \mathbf{1}_n\rangle = 1\}$, $\psi_0(v) = \langle v, \ln v\rangle$.[3]

- least squares with ℓ_1 regularization: (3.23) with $g_i(v) = \frac{1}{100}\left\|v\right\|_1$ for all $i \in \mathcal{V}$, $\mathbb{X}_0 = \mathbb{R}^n$, $\psi_0(v) = \frac{1}{2}\left\|v\right\|_2^2$.

For (3.19) and (3.24)), we set $r = \frac{1}{10}\mathbf{1}_{|\mathcal{E}|}$, $l = (\beta + \lambda)r$ and $\alpha^k \equiv \frac{1}{\beta+\lambda}$; for distributed mirror descent, we use step size $\frac{1}{\sqrt{k}}$ and doubly stochastic matrix $P = I - \frac{1}{1+\Delta}L(\mathcal{G})$, where $L(\mathcal{G}) = E(\mathcal{G})E(\mathcal{G})^\top$ and Δ is the largest diagonal element of $L(\mathcal{G})$; for mirror-prox method, we use step size $\min\{\frac{1}{2\lambda}, \frac{r}{2\sqrt{l\sigma}}\}$ for problem (3.6) and (3.23); see [26, Thm. 5.2]. In the noisy gradient case, we sample η^k from Gaussian distribution $\mathcal{N}(0, \sigma I_{|\mathcal{V}|n})$ for all k.

[3] $\ln v$ denotes the element-wise natural logarithm of v.

The convergence of algorithms are shown in Fig. 3.4 and Fig. 3.5. We can see that the convergence of (3.12) and (3.24) behaves no worse than mirror-prox method [128] and its composite objective extension [76] using only half of the computation cost per iteration; they all reach a “noise floor" when gradient are corrupted by noise. In the ℓ_1 regularized least squares case, both mirror-prox and RLC significantly outperform distributed mirror descent method since they use the shrinkage operator rather than subgradients of ℓ_1 norm.

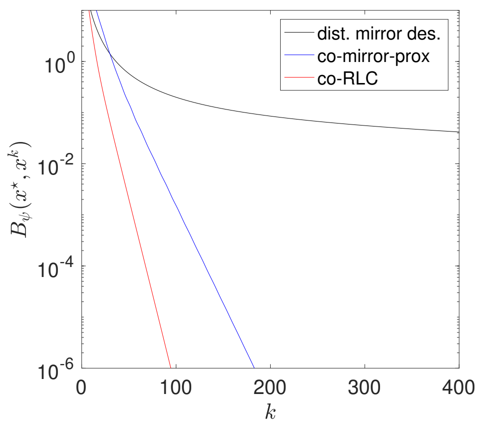


(a) $\gamma = 0$ (b) $\gamma = 10^{-3}$

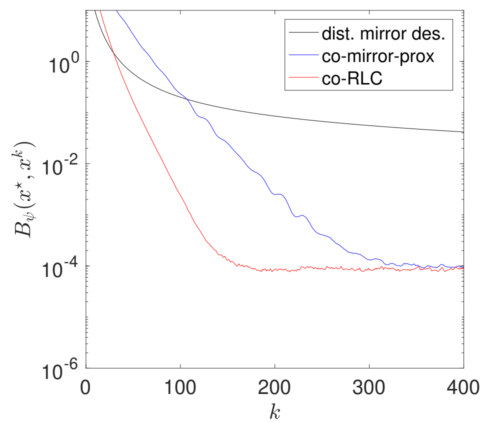
Figure 3.4: Least squares over simplex.

3.2 Trajectory optimization

In this section, we consider finite-horizon discrete-time convex trajectory optimization problem in optimal control problems, which plays an vital role in model predictive control (MPC). Perhaps surprisingly, such trajectory optimization is actually closely related to the distributed optimization problem (3.6). As a result, the algorithm we designed for distributed optimization can be seamlessly generalized to trajectory optimization.



(a) $\sigma = 0$



(b) $\sigma = 10^{-3}$

Figure 3.5: ℓ_1 regularized least squares.

Throughout this section we consider the following optimization problem

$$\begin{array}{ll} \underset{z}{\text{minimize}} & \frac{1}{2}z^\top H z + h^\top z \\ \text{subject to} & Gz = g, \;\; z \in \mathbb{Z}, \end{array} \tag{3.29}$$

where the trajectory variable z aims to minimize a convex quadratic cost function $\frac{1}{2}z^\top H z + h^\top z$ subject to linear dynamics constraints $Gz = g$ together with convex state and input constraint $z \in \mathbb{Z}$. Throughout we assume $\mathbb{Z}$ is the Cartesian product of convex sets whose Euclidean projection can be evaluated at low computational cost. Such assumption applies to many popular state and input constraints used in MPC; see Tab. 3.3 for some examples (where we assume function f is continuous, convex and finite valued) and [7] for a detailed discussion. Notice that optimization (3.1) is a special case of (3.29) with the following choice of parameters

$$
\begin{aligned}
z =&[u_1^\top, x_2^\top, \ldots, u_{L-1}^\top, x_L^\top]^\top,\;\; \mathbb{Z} = \textstyle\prod_{t=1}^{L-1}(\mathbb{U}_t \times \mathbb{X}_{t+1}),\;\; H = \mathbf{blkdiag}(R_1, Q_2, \ldots, R_{L-1}, Q_L),\\
h =&[0^\top, -y_2^\top Q_2, \ldots, 0^\top, -y_L^\top Q_L]^\top,\;\; g = [x_1^\top A^\top, 0^\top, \cdots, 0^\top]^\top.\\
G =& \begin{bmatrix} -B & I & & & & & \\ & -A & -B & I & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & -A & -B & I \end{bmatrix}
\end{aligned}
\tag{3.30}
$$

Table 3.3: Examples of simple convex sets and projections

Set $\mathbb{X}$	Projection of x onto $\mathbb{X}$ if $x \notin \mathbb{X}$
$\{x \mid \|x\|_2 \leqslant \alpha\}$	$\frac{\alpha}{\|x\|_2} x$
$\{x \mid l \leqslant x \leqslant u\}$	$\min\{\max\{x, l\}, u\}$
$\{x \mid \langle a, x\rangle \leqslant \alpha, a \neq 0\}$	$x - (\langle a, x\rangle - \alpha)\frac{a}{\|a\|_2}$
$\{x = (y, \alpha) \mid \|y\|_2 \leqslant \alpha\}$	$(0, 0)$ if $\|y\|_2 \leqslant -\alpha$; $\frac{\|y\|_2+\alpha}{2\|y\|_2}(y, \|y\|_2)$ otherwise
$\{x = (y, \alpha) \mid f(y) \leqslant \alpha\}$	$(x, f(x))$ where x solves $y \in x + (f(x) - \alpha)\partial f(x)$
$\{x \mid f(x) \leqslant \alpha\}$	$(I + \mu\partial f)^{-1}(x)$ where μ solves $f((I + \mu\partial f)^{-1}(x)) = \alpha$

Inspired by the method (3.12), we propose to solve (3.29) using the following iterations

$$
z^{k+1} = \pi_{\mathbb{Z}}[z^k - \alpha^k(Hz^k + h + G^\top w^k + \beta^k G^\top(Gz^k - g))], \tag{3.31a}
$$

$$
w^{k+1} = w^k + \beta^k(Gz^{k+1} - g). \tag{3.31b}
$$

We now prove the convergence properties of method (3.31). First, we group our assumptions as follows.

Assumption 3.3. *Suppose*

1. *set* $\mathbb{Z} \subset \mathbb{R}^n$ *is closed and convex; matrix* $H \in \mathbb{R}^{n\times n}$ *is symmetric, matrix* $G \in \mathbb{R}^{m\times n}$ *has full row rank, there exists* $0 \leqslant \mu \leqslant \lambda$ *and* $\sigma \geqslant 0$ *such that* $\mu I \preceq H \preceq \lambda I$ *and* $G^\top G \preceq \sigma I$.

2. *there exists* $z^\star \in \mathbb{R}^n$ *and* $w^\star \in \mathbb{R}^m$ *such that*

$$Gz^\star = g, \quad z^\star \in \mathbb{Z}, \tag{3.32a}$$

$$\langle Hz^\star + h + G^\top w^\star, z - z^\star \rangle \geqslant 0, \quad \forall z \in \mathbb{Z}. \tag{3.32b}$$

Equation (3.32) gives the Karush–Kuhn–Tucker conditions of problem (3.29). Under the Slater condition for equalities, equation (3.32) holds if and only if $z^\star$ is an optimal solution for problem (3.29); see [131, Thm.3.1.27].

The following lemma shows the key property of two consecutive iterations generated by method (3.31).

Lemma 3.4. *Suppose Assumption 3.3 holds and sequence* $\{z^k, w^k\}$ *is generated by* (3.31). *If* $\lambda + \sigma\beta^k = \frac{1}{\alpha^k}$ *for all* $k \geqslant 1$, *then*

$$\begin{aligned}
&\frac{\beta^k}{2}\left\|Gz^k - g\right\|_2^2 + \frac{1}{2}\left\|z^{k+1} - z^\star\right\|_H^2 \\
\leqslant &\frac{1}{2}(\frac{1}{\alpha^k} - \mu)\left\|z^k - z^\star\right\|_2^2 + \frac{1}{2\beta^k}\left\|w^k - w^\star\right\|_2^2 - \frac{1}{2\alpha^k}\left\|z^{k+1} - z^\star\right\|_2^2 - \frac{1}{2\beta^k}\left\|w^{k+1} - w^\star\right\|_2^2.
\end{aligned}$$

Proof. Step 1, applying Lemma A.3 to (3.31a) gives

$$\begin{aligned}
0 \leqslant &\frac{1}{\alpha^k}\langle z^{k+1} - z^k, z^\star - z^{k+1}\rangle + \beta^k\langle Gz^k - g, g - Gz^{k+1}\rangle \\
&+ \langle Hz^k + h, z^\star - z^{k+1}\rangle + \langle w^k, G(z^\star - z^{k+1})\rangle,
\end{aligned} \tag{3.33}$$

where we also used (3.32a). Next, (3.32b) implies that

$$0 \leqslant -\langle Hz^\star + h, z^\star - z^{k+1}\rangle - \langle w^\star, G(z^\star - z^{k+1})\rangle. \tag{3.34}$$

Step 2, one can directly verify the following four identities, which can be interpreted as instances of the *law of cosines*; see Fig. 3.6 for an illustration.

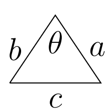
$$\langle z^{k+1} - z^k, z^\star - z^{k+1}\rangle = \frac{1}{2}\left\|z^k - z^\star\right\|_2^2 - \frac{1}{2}\left\|z^{k+1} - z^\star\right\|_2^2 - \frac{1}{2}\left\|z^{k+1} - z^k\right\|_2^2, \tag{3.35}$$

$$\langle Gz^k - g, g - Gz^{k+1}\rangle = \frac{1}{2}\left\|G(z^{k+1} - z^k)\right\|_2^2 - \frac{1}{2}\left\|Gz^k - g\right\|_2^2 - \frac{1}{2}\left\|Gz^{k+1} - g\right\|_2^2, \tag{3.36}$$

$$\langle H^{\frac{1}{2}}(z^k - z^\star), H^{\frac{1}{2}}(z^\star - z^{k+1})\rangle = \frac{1}{2}\left\|z^{k+1} - z^k\right\|_H^2 - \frac{1}{2}\left\|z^k - z^\star\right\|_H^2 - \frac{1}{2}\left\|z^{k+1} - z^\star\right\|_H^2, \tag{3.37}$$

$$\begin{aligned}&\frac{1}{2}\left\|w^{k+1} - w^\star\right\|_2^2 - \frac{1}{2}\left\|w^k - w^\star\right\|_2^2\\ =&\langle w^k - w^\star, w^{k+1} - w^k\rangle + \frac{1}{2}\left\|w^{k+1} - w^k\right\|_2^2 = \beta^k\langle w^k - w^\star, G(z^{k+1} - z^\star)\rangle + \frac{(\beta^k)^2}{2}\left\|Gz^{k+1} - g\right\|_2^2,\end{aligned} \tag{3.38}$$

where matrix $H^{\frac{1}{2}}$ in (3.37) is the positive semi-definite square root of H, and the last step in (3.38) is due to (3.31b) and (3.32a).



$$-ab\cos\theta = \tfrac{1}{2}c^2 - \tfrac{1}{2}a^2 - \tfrac{1}{2}b^2$$

Figure 3.6: The law of cosines.

Step 3, the assumption that $0 \preceq \mu I \preceq H \preceq \lambda I$ and $G^\top G \preceq \sigma I$ implies the following

$$\frac{\mu}{2}\left\|z^k - z^\star\right\|_2^2 \leqslant \frac{1}{2}\left\|z^k - z^\star\right\|_H^2, \tag{3.39a}$$

$$\frac{1}{2}\left\|z^{k+1} - z^k\right\|_H^2 \leqslant \frac{\lambda}{2}\left\|z^{k+1} - z^k\right\|_2^2, \tag{3.39b}$$

$$\frac{1}{2}\left\|G(z^{k+1} - z^k)\right\|_2^2 \leqslant \frac{\sigma}{2}\left\|z^{k+1} - z^k\right\|_2^2. \tag{3.39c}$$

Finally, summing up together (3.33), (3.34), $\frac{1}{\alpha^k}\times$(3.35), $\beta^k\times$(3.36), (3.37), $\frac{1}{\beta^k}\times$(3.38), (3.39a), (3.39b) and $\beta^k\times$(3.39c), then using the assumption that $\lambda+\sigma\beta^k = \frac{1}{\alpha^k}$, we obtain the desired result.

□

We start with the case where $\mu = 0$, *i.e.*, matrix H is only positive semi-definite and the objective function in problem (3.29) is only convex. The following theorem shows that, using constant step sizes, the iterations in (3.31) converge to optimum at the rate of $O(1/k)$.

Theorem 3.5. *Suppose Assumption 3.3 hold with $\mu = 0$, and sequence $\{v^k, z^k, w^k\}$ is generated by* (3.31) *with $\alpha^k = \frac{1}{\beta\sigma+\lambda}$ and $\beta^k = \beta$ for some $\beta > 0$ and all $k \geqslant 1$. Let $V^1 = \frac{1}{2\alpha}\left\|z^1 - z^\star\right\|_2^2 + \frac{1}{2\beta}\left\|w^1 - w^\star\right\|_2^2$, then*

$$\frac{1}{2}\left\|G\hat{z}^k - g\right\|_2^2 \leqslant \frac{1}{\beta k}V^1, \quad \frac{1}{2}\left\|\tilde{z}^k - z^\star\right\|_H^2 \leqslant \frac{1}{k}V^1.$$

where $\hat{z}^k = \frac{1}{k}\sum_{j=1}^k z^j$ and $\tilde{z}^k = \frac{1}{k}\sum_{j=1}^k z^{j+1}$.

Proof. With this choice of α^k and β^k, the inequality in Lemma 3.4 becomes the following: for all $j \geqslant 1$,

$$\frac{\beta}{2}\left\|Gz^j - g\right\|_2^2 + \frac{1}{2}\left\|z^{j+1} - z^\star\right\|_H^2 \leqslant V^j - V^{j+1},$$

where $V^j = \frac{1}{2\alpha}\left\|z^j - z^\star\right\|_2^2 + \frac{1}{2\beta}\left\|w^j - w^\star\right\|_2^2$. Summing up this inequality for $j = 1, \ldots, k$ gives

$$\sum_{j=1}^k \Big(\frac{\beta}{2}\left\|Gz^j - g\right\|_2^2 + \frac{1}{2}\left\|z^{j+1} - z^\star\right\|_H^2\Big) \leqslant V^1 - V^{k+1} \leqslant V^1$$

where the last step is because $V^{k+1} \geqslant 0$. Hence

$$\frac{\beta}{2}\sum_{j=1}^k \left\|Gz^j - g\right\|_2^2 \leqslant V^1, \quad \frac{1}{2}\sum_{j=1}^k \left\|z^{j+1} - z^\star\right\|_H^2 \leqslant V^1.$$

Finally, applying Jensen's inequality (A.4) to the above two inequalities gives the desired results. □

Theorem 3.6. *Suppose Assumption 3.3 hold with $\mu > 0$, and sequence $\{v^k, z^k, w^k\}$ is generated by (3.31) with $\alpha^k = \frac{2}{(k+1)\mu+2\lambda}$, $\beta^k = \frac{(k+1)\mu}{2\sigma}$ for all $k \geqslant 1$. Let $V^1 = \frac{1}{2(\mu+\lambda)}\left\|z^1 - z^\star\right\|_2^2 + \frac{\sigma}{2\mu}\left\|w^1 - w^\star\right\|_2^2$, then*

$$
\begin{aligned}
\frac{1}{2}\left\|G\hat{z}^k - g\right\|_2^2 \leqslant & \frac{12\lambda\sigma}{\mu^2 k(k^2+6k+11)} V^1, \\
\frac{1}{2}\left\|\tilde{z}^k - z^\star\right\|_H^2 \leqslant & \frac{4\lambda}{\mu k(k+5)} V^1,
\end{aligned}
$$

where $\hat{z}^k = \frac{3}{k(k^2+6k+11)}\sum_{j=1}^k (j+1)(j+2)z^j$ and $\tilde{z}^k = \frac{2}{k(k+5)}\sum_{j=1}^k (j+2)z^{j+1}$.

Proof. With this choice of α^k and β^k, the inequality in Lemma 3.4 becomes the following: for all $j \geqslant 1$,

$$
\frac{\beta^j}{2}\left\|Gz^j - g\right\|_2^2 + \frac{1}{2}\left\|z^{j+1} - z^\star\right\|_H^2 \leqslant \frac{1}{2}\left(\frac{1}{\alpha^j} - \mu\right)\left\|z^j - z^\star\right\|_2^2 + \frac{1}{2\beta^j}\left\|w^j - w^\star\right\|_2^2 - V^{j+1}, \quad (3.40)
$$

where $V^j = \frac{1}{2\alpha^{j-1}}\left\|z^j - z^\star\right\|_2^2 + \frac{1}{2\beta^{j-1}}\left\|w^j - w^\star\right\|_2^2$. Let $\kappa = \lambda/\mu \geqslant 1$, then it is straightforward to verify the following

$$
\begin{aligned}
\left(\frac{1}{\alpha^j} - \mu\right)(j + 2\kappa) = & \frac{1}{\alpha^{j-1}}(j + 2\kappa - 1), \\
\frac{1}{\beta^j}(j + 2\kappa) \leqslant & \frac{1}{\beta^{j-1}}(j + 2\kappa - 1).
\end{aligned} \quad (3.41)
$$

Hence multiplying (3.40) with $(j + 2\kappa)$ and substituting in (3.41) we can show

$$
\frac{(j+1)(j+2\kappa)\mu}{4\sigma}\left\|Gz^j - g\right\|_2^2 + \frac{j+2\kappa}{2}\left\|z^{j+1} - z^\star\right\|_H^2 \leqslant (j + 2\kappa - 1)V^j - (j + 2\kappa)V^{j+1}.
$$

Summing up this inequality for $j = 1, 2, \ldots, k$ gives

$$
\sum_{j=1}^k \left(\frac{(j+1)(j+2\kappa)\mu}{4\sigma}\left\|Gz^j - g\right\|_2^2 + \frac{j+2\kappa}{2}\left\|z^{j+1} - z^\star\right\|_H^2\right) \leqslant 2\kappa V^1 - (k + 2\kappa)V^{k+1} \leqslant 2\kappa V^1.
$$

where the last step is because $V^{k+1} \geqslant 0$. Since $\kappa \geqslant 1$, the above inequality implies the following

$$\sum_{j=1}^{k} \frac{(j+1)(j+2)\mu}{4\sigma} \left\|Gz^j - g\right\|_2^2 \leqslant 2\kappa V^1,$$

$$\sum_{j=1}^{k} \frac{j+2}{2} \left\|z^{j+1} - z^\star\right\|_H^2 \leqslant 2\kappa V^1.$$

Finally, applying Jensen's inequality (A.4) to the above two inequalities and using $\kappa = \lambda/\mu$ gives the desired results. □

Compared with Theorem 3.5, Theorem 3.6 used an averaged sequence with increasing weights, similar to those in subgradient method [97] and accelerated ADMM [186]. Notice that Theorem 3.6 shows that the constraint violation converges at a $O(1/k^3)$ rate, even faster than $O(1/k^2)$, which is highly desirable in practice as constraint violation is often used as the stopping criterion.

We are now ready to implement (3.31) for problem (3.1). We partition variables w and v as follows

$$v = [v_1^\top, v_2, \ldots, v_T^\top]^\top, \;\; w = [w_1^\top, w_2, \ldots, w_T^\top]^\top, \tag{3.42}$$

where $v_t, w_t \in \mathbb{R}^{n_x}$ corresponds to constraint $x_t = Ax_{t-1} + Bu_{t-1}$ for $1 \leqslant t \leqslant T$. In addition, the separable structure of set $\mathbb{Z}$ defined by (3.30) allows separable computation of its Euclidean projection. Based on these observations, we implement algorithm (3.31) for problem (3.1) in Algorithm 9, where we introduce dummy parameters $A_T v_{T+1} \equiv 0$ to simplify our notation. Notice that updates of variables corresponding to different value of t can be executed in parallel, hence the algorithm run-time can be almost independent of horizon T.

We finish this section by comparing our method against several existing methods over a trajectory-planning problem with keep-out-zone constraints, where all parameters are chosen as unit-less for simplicity.

We consider a trajectory-planning (finite horizon optimal control) problem where the goal is to track a beeline trajectory from initial to target position while avoiding collision with

Algorithm 9 PI projected gradient method

Input: x_0; $\mathbb{X}_t, \mathbb{U}_{t-1}$, Q_t, y_t, R_{t-1}, A, B for all $1 \leqslant t \leqslant T$. Initialize $k = 1$, u_{t-1}, x_t, w_t for all $1 \leqslant t \leqslant T$; let $A_T v_{T+1} \equiv 0$.
  while $k \leqslant k_{\max}$ **do**
    $k \leftarrow k + 1$
    For all $1 \leqslant t \leqslant T$:
    $v_t \leftarrow w_t + \beta^k(x_t - Ax_{t-1} - Bu_{t-1})$
    $u_{t-1} \leftarrow \pi_{\mathbb{U}_{t-1}}[u_{t-1} - \alpha^k(R_{t-1}u_{t-1} - B^\top v_t)]$
    $x_t \leftarrow \pi_{\mathbb{X}_t}[x_t - \alpha^k(Q_t(x_t - y_t) + v_t - A_t^\top v_{t+1})]$
    $w_t \leftarrow w_t + \beta^k(x_t - Ax_{t-1} - Bu_{t-1})$
  end while

Output: $\{u_0, x_1, \ldots, u_{T-1}, x_T\}$

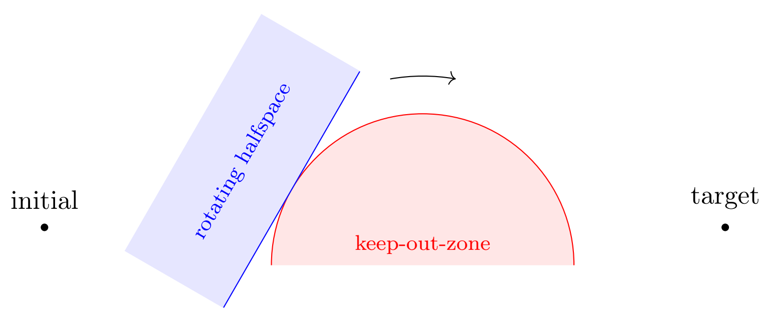


Figure 3.7: Trajectory-planning with rotating halfspace constraint.

a circular keep-out-zone; see Fig. 3.7 for an illustration. Here, this problem is an instant of an MPC problem, which is solved repetitively as new state information becomes available. The dynamics of the system is modeled as a double integrator with sampling time 0.5 s. The system is subject to ℓ_2 norm constraints on its velocity $q \in \mathbb{R}^2$ and acceleration input $u \in \mathbb{R}^2$. In addition, a rotating half-space constraint is imposed on its position $p \in \mathbb{R}^2$, which convexifies the keep-out-zone; see [196] for a detailed discussion. We model this tracking problem as a special case of problem (3.1) with the following choice of parameters:

$$A = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \\ 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \tag{3.43a}$$

$$Q_t = \mathbf{diag}(1, 0.5, 1, 0.5), \;\; R_{t-1} = \mathbf{diag}(1, 0.5), \tag{3.43b}$$

$$\mathbb{X}_t = \left\{ x = \begin{bmatrix} p \\ q \end{bmatrix} \;\middle|\; \begin{bmatrix} -\cos(\theta t) \\ \sin(\theta t) \end{bmatrix}^\top p \geqslant 2, \; \|q\|_2 \leqslant 0.25 \right\}, \tag{3.43c}$$

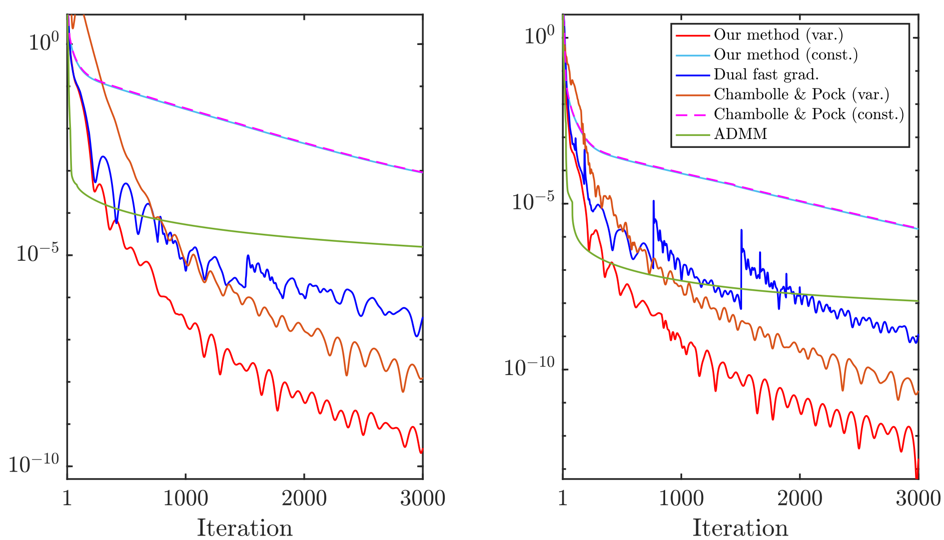
$$\mathbb{U}_{t-1} = \{u | \, \|u\|_2 \leqslant 0.1\}, \;\; x_0 = \begin{bmatrix} -2.5 & 0.6 & 0 & 0 \end{bmatrix}^\top, \tag{3.43d}$$

for $1 \leqslant t \leqslant T$, where $\theta = 0.063$ in (3.43c) is a constant rotation rate [196]. Note that Q_t and R_t in (3.43b) are diagonal but not identity, which is common in practice. The reference trajectory $\{y_t\}_{t=1}^T$ in (3.1) is chosen as a beeline trajectory from initial position $(-2.5, 0.6)$ to target position $(2.9, 0.3)$ without considering the position constraint on p in (3.43c).

We compare our method against several popular existing methods. In terms of step sizes: for our method (3.31), we choose α^k and β^k according to Theorem 3.5 and Theorem 3.6 for constant and, respectively, varying step sizes; for dual fast gradient method, we choose α according to [148, Thm.1]; for Chambolle & Pock method, we choose the constant step sizes in (3.58) according to [35, Rem. 1], and the varying step sizes in (3.59) according to the "optimal rule" in [35, Sec. 5.2]; for ADMM, we choose $\alpha = 2$ as suggested in [84]. In addition, the inner loop iterations used by each iteration of method (3.52) are warm-started using results from the last outer iteration and terminated if $\|z^{j+1} - z^j\|_2 / \|z^j\|_2 \leqslant \epsilon_{\text{inner}}$, where ϵ_{inner} is chosen between 0.1% and 0.01%.

We summarize our results as follows. Fig. 3.8 shows the convergence over iterations of different algorithms with same initialization for $T = 25$, where $z^\star$ is computed using ECOS [47] together with JuMP [52]. Fig. 3.9 shows the computation costed by different algorithms for $T = \{5, 15, 25, 35, 45\}$ to reach the tolerance for constraint violation (we use ℓ_∞-norm since it measures the maximum pointwise constraint violation along the trajectory), where each data point is averaged over 200 independent experiments using initialization sampled from standard normal distribution. Note that we omitted method (3.58) and (3.31) with constant step sizes in Fig. 3.9b due to their slow convergence.

In these simulations, our method with varying step sizes (var.) outperforms the others. Our method with constant step sizes (const.) converges slower, but almost identically to Chambolle & Pock method with constant step sizes in (3.58), since they do not exploit the strong convexity of the objective functions.

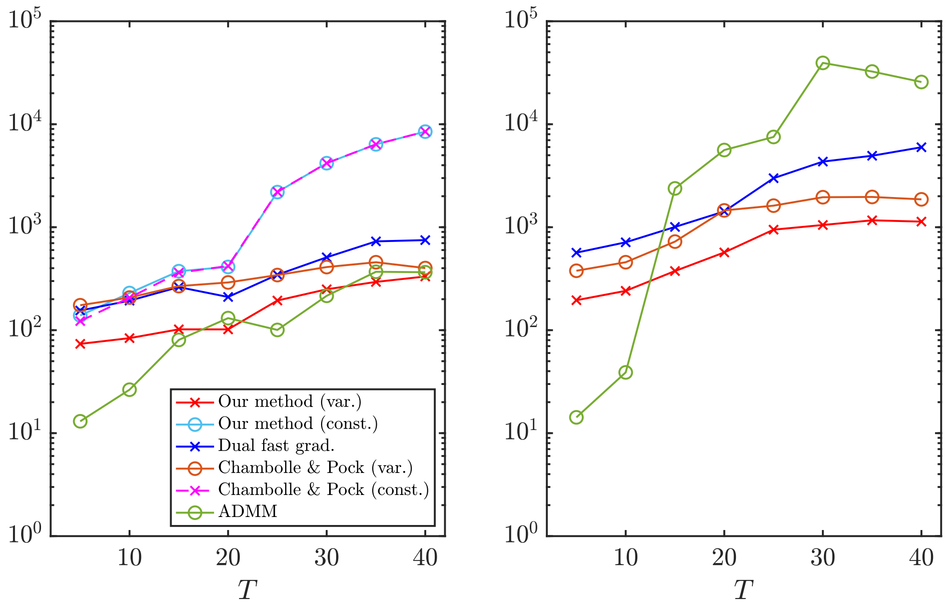


(a) $\|z - z^\star\|_2^2$ (b) $\|Gz - g\|_2^2$

Figure 3.8: Convergence over iterations for $T = 25$.

3.3 Related work and remarks: distributed optimization

In this section, we briefly review some of the existing work on distributed optimization that most relevant to the results in §3.1.



(a) $\epsilon = 10^{-3}$ (b) $\epsilon = 10^{-5}$

Figure 3.9: Number of projection $\pi_{\mathbb{Z}}[\cdot]$ costed to reach condition $\|Gz - g\|_\infty \leqslant \epsilon$. Each data point is averaged over 200 simulations using random initialization.

3.3.1 Distributed subgradient method and its variations

One of the earliest work on distributed optimization algorithm is the distributed subgradient method proposed by [125], which solves the special case of (B.1) with $\mathbb{X}_0 = \mathbb{R}^n$ using the following iterations

$$y^k = (I - \beta E(\mathcal{G})E(\mathcal{G})^\top)x^k \tag{3.44a}$$

$$x^{k+1} = y^k - \alpha^k u^k, \;\; u^k \in \partial f(y^k). \tag{3.44b}$$

where $\beta > 0$ is chosen such that the second largest eigenvalues of $(I - \beta E(\mathcal{G})E(\mathcal{G})^\top)$ lie in open interval $(0, 1)$ [4]. The convergence proof in [125] analyze the behavior of (3.44) as a linear dynamic system with disturbance, where the disturbance corresponds to the subgradients

[4]In general, one can replace $(I - \beta E(\mathcal{G})E(\mathcal{G})^\top)$ with a doubly stochastic matrix M such that $M_{ij} > 0$ only if $\{i, j\}$ is an edge in $\mathcal{G}$

with bounded norm. This work was further extended to distributed projected subgradient method [127] to handle the case where $\mathbb{X}_0$ is a general closed convex set. The resulting algorithm iterate as follows

$$
\begin{aligned}
y^k =& (I - \beta E(\mathcal{G})E(\mathcal{G})^\top)x^k && (3.45a)\\
x^{k+1} =& \pi_{\mathbb{X}}[y^k - \alpha^k u^k], \;\; u^k \in \partial f(y^k). && (3.45b)
\end{aligned}
$$

Later, algorithm 3.45 was further extended to distributed mirror descent method [50, 183, 183, 104, 176, 105, 46], which takes the following form

$$
\begin{aligned}
y^{k+1} =& (I - \beta E(\mathcal{G})E(\mathcal{G})^\top)x^k && (3.46a)\\
x^{k+1} =& \operatorname*{argmin}_{x \in \mathbb{X}} \alpha^k \langle u^k, x\rangle + B_\psi(x, y^k), \;\; u^k \in \partial f(y^k). && (3.46b)
\end{aligned}
$$

The difference between the convergence analysis of distributed projected subgradient method in (3.45) and the distributed mirror descent method in (3.46) is similar to the difference between the one for projected subgradient method and the one for mirror descent; see Appendix B.2 and Appendix B.4 for details.

3.3.2 Distributed alternating direction method of multipliers

Along a different line of research is the distributed alternating direction method of multiplier (ADMM) [74, 44, 77]. These methods are inspired by ADMM, which alternatively optimizes an augmented Lagrangian with respect to splitted primal variables and dual variables. [25], The main challenges of distributed ADMM is to find a separable approximation to the coupled quadratic penalty term in augmented Lagrangian. In particular, a Gauss-Seidel approximation [74, 77] was proposed in [178], which results in sequential updates on the vertices. On the other hand, a Jacobian approximation based variant of ADMM [83, 44] allows simultaneous updates [172]. One of the first Jacobian approximation based distributed

ADMM was introduced in [121], which iterates as follows.

$$x^{k+1} = \underset{x\in\mathbb{X}}{\operatorname{argmin}}\ \alpha\langle\sqrt{l}E(\mathcal{G})u^k + rE(\mathcal{G})E(\mathcal{G})^\top x^k, x\rangle + \alpha f(x) + \frac{1}{2}\left\|x - x^k\right\|_2^2 \tag{3.47a}$$

$$u^{k+1} = u^k + rE(\mathcal{G})^\top x^{k+1} \tag{3.47b}$$

In [191], Yu *et al.* tried to extended the quadratic function used in (3.47a) to general Bregman divergence. The resulting algorithm used an “mirror averaging" step that generalizes the traditional averaging step (*e.g.*, (3.46a)): instead of an average based on quadratic distance, the mirror averaging step in [191] uses an average based on Bregman divergence. The convergence analysis in [191] follow similar steps as those in [171]. The results in [191] lays one of the foundation of the results in § 3.1.2.

3.3.3 Distributed optimization methods via discretizing ODEs

Another motivation of results in § 3.1.2 is the close connection between first order optimiztaion algorithm and discretization of nonlinear ordinary differential equations [173]. One of the most influential work in this direction is the one by Su *et al.* [161], followed by its extension to cases where distance is measured by Bregman divergence rather than quadratic functions [95, 180]. Recently the approximate duality gap theory [45] provided a unified perspective of these research; we refer the readers to Appendix B.6 for some relevant details[5]. A particularly result relevant to distributed optimization is the mirror-prox method [128] designed for general convex-concave saddle-point problem. When applied to the distributed optimization problem (3.6), the mirror-prox method iterates as follows

$$y^k = \underset{y\in\mathbb{X}}{\operatorname{argmin}}\ \alpha\langle\sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k), y\rangle + B_\psi(y, x^k), \tag{3.48a}$$

[5]All these work were presented at the University of Washington, Seattle. In particular, Walid Krichene presented the work in [94, 93] in one of 2018 CORE seminar. Michael I. Jordan presented the work in [180] at 2018 Taskar Memorial Distinguished Lecturer. Jelena Diakonikolas presented her work in [45, 37] at 2019 ADSI summer school lectures.

$$v^k = u^k + \alpha\sqrt{l}E(\mathcal{G})^\top x^k, \tag{3.48b}$$

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{X}} \ \alpha\langle\sqrt{l}E(\mathcal{G})v^k + \nabla f(y^k), x\rangle + B_\psi(x, x^k), \tag{3.48c}$$

$$u^{k+1} = u^k + \alpha\sqrt{l}E(\mathcal{G})^\top y^k. \tag{3.48d}$$

Using steps similar to those in § 3.1.2, one can show that mathod (3.48) is a discretization of the following ODE

$$\begin{aligned} \frac{d}{dt}\nabla\psi(x(t)) &= -\sqrt{l}E(\mathcal{G})u(t) - \nabla f(x(t)), \\ \frac{d}{dt}u(t) &= \sqrt{l}E(\mathcal{G})^\top x(t), \end{aligned} \tag{3.49}$$

In particular, the (y^k, v^k) is the "prediction value" obtained using Euler-forward scheme, which is used later to compute (x^{k+1}, u^{k+1}), the "correction value". Motivated by this work, Yu *et al.* [192] propose equation (3.9) as a continuous-time algorithm for problem (3.6), together with the following discrete time algorithm based on Euler-backward discretization

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{X}} \ \alpha\langle rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k, x\rangle + \alpha f(x) + B_\psi(x, x^k), \tag{3.50a}$$

$$u^{k+1} = u^k + \alpha\sqrt{l}E(\mathcal{G})^\top x^{k+1}. \tag{3.50b}$$

The main challenge of implementing the algorithm in (3.50) is that the minimization in (3.50a) in general requires an iterative method itself, which is a common drawback of all ADMM-based methods. As a alternative solution, an algorithm based on Euler-forward discretization of equation (3.9) was introduced in [190], which modifies the iterations in (3.50) as follows

$$x^{k+1} = \operatorname*{argmin}_{x\in\mathbb{X}} \ \alpha\langle rE(\mathcal{G})E(\mathcal{G})^\top x^k + \sqrt{l}E(\mathcal{G})u^k + \nabla f(x^k), x\rangle + B_\psi(x, x^k), \tag{3.51a}$$

$$u^{k+1} = u^k + \alpha\sqrt{l}E(\mathcal{G})^\top x^{k+1}. \tag{3.51b}$$

Similar to the one in distributed mirror descent method used in (3.46), the minimization step in (3.51a) is no different from that in mirror descent method (see Appendix B.4), which often allows explicit solution rather than requiring an iterative method itself. The results on method (3.51) along with its extensions are all included in § 3.1.2.

Notice all the aforementioned distributed mirror descent methods are based on discretization of different ODEs using different schemes. For example, step (3.46a) in (3.46) is essentially a discretization of ODE (3.4) using Euler-forward scheme. Method (3.48) is a discretization of ODE (3.49) using Euler-forward scheme. Finally, method (3.50) and (3.12) are discretization of ODE (3.9) using Euler-backward[6] and, respectively, Euler-forward scheme. Based on these observations, we compare all the aforementioned methods in Table 3.4, in terms of their assumptions on the objective function, the ODE and discretization schemes they use, and their convergence rate. Notice that method (3.44) and (3.45) are both special case of (3.46), and similarly (3.47) is a special case of (3.50), so we omit them in Table 3.4.

Algorithm	Objective function	ODE	Discretization	Convergence
[50, 183, 104, 176, 105, 46]	non-smooth, convex	(3.4)	Euler-forward	$\mathcal{O}(1/\sqrt{k})$
[195]	non-smooth, strongly convex	(3.4)	Euler-forward	$\mathcal{O}(1/k)$
[128, 86, 76]	smooth, convex	(3.49)	predictor-corrector	$\mathcal{O}(1/k)$
[192]	non-smooth, convex	(3.9)	Euler-backward	$\mathcal{O}(1/k)$
[190]	smooth, convex	(3.9)	Euler-forward	$\mathcal{O}(1/k)$

Table 3.4: Different distributed mirror descent methods.

3.4 *Related work and remarks: trajectory optimization*

The trajectory optimization problem (3.29) has been studied as the key ingradient of model predictive control [120, 119, 142, 91, 55]. In addition its solution is also an integral part of problems with nonlinear dynamics and non-convex constraints. In these cases, a sequence of convex sub-problems modeled by (3.29) are solved to obtain the solution of the original non-convex problem, as done in sequential convex programming [64, 22] and successive

[6] We call it Euler-backward scheme because x^{k+1} is defined as an implicit function in (3.50a)

convexification methods [112, 111, 113]. Many first order primal-dual methods have been developed for problem (3.29). In this section, we briefly review some of them.

3.4.1 Dual fast gradient method

The dual fast gradient method [148, 138, 66] solves problem (3.29) by applying the Nesterov's accelerated gradient method (see Appendix B.6) to the dual problem of (3.29). Such method requires matrix H to be positive definite, and iterates as follows

$$z^{k+1} = \operatorname*{argmin}_{z \in \mathbb{Z}} \tfrac{1}{2} z^\top H z + h^\top z + \langle v^k, Gz \rangle, \tag{3.52a}$$

$$w^{k+1} = v^k + \alpha(Gz^{k+1} - g), \tag{3.52b}$$

$$v^{k+1} = w^{k+1} + \tfrac{k}{k+3}(w^{k+1} - w^k). \tag{3.52c}$$

In general, the minimization step in (3.52a) can only be solved approximately using another inner loop of Nesterov's accelerated gradient method [88], which iterates as follows [131, Sec. 2.2] (j denotes the inner loop iteration counter)

$$\begin{aligned} z^{j+1} &= \pi_{\mathbb{Z}}[y^j - \tfrac{1}{\lambda}(Hy^j + h + G^\top v^k)], \\ y^{j+1} &= z^{j+1} + \tfrac{\sqrt{\lambda}-\sqrt{\mu}}{\sqrt{\lambda}+\sqrt{\mu}}(z^{j+1} - z^j), \end{aligned} \tag{3.53}$$

where $0 \prec \mu I \preceq H \preceq \lambda I$. This "double-loop" algorithm have been implemented as code-generation tool in FiOrdOs [165] and the μAO-MPC [198] package.

3.4.2 Alternating direction method of multipliers

Another popular method to solve optimization (3.29) is the alternating direction method of multipliers (ADMM) [134, 84, 160, 40]. Among its many variations, here we use the form developed in [84] as an example

$$y^{k+1} = \operatorname*{argmin}_{z : Gz = g} \tfrac{1}{2} z^\top H z + h^\top z + \tfrac{1}{2\alpha} \left\| z + w^k - y^k \right\|_2^2, \tag{3.54a}$$

$$z^{k+1} = \pi_{\mathbb{Z}}[y^{k+1} + w^k], \tag{3.54b}$$

$$w^k = w^k + z^{k+1} - y^{k+1}. \tag{3.54c}$$

Notice that ADMM solves two subproblems for primal variables: minimization of a quadratic function over a hyperplane in (3.54a) and the projection in (3.54b). The minimization in (3.54a) is equivalent to solving the following system of linear equations for variable z

$$\begin{bmatrix} H + \frac{1}{\alpha}I & G^\top \\ G & 0 \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} -h - \frac{1}{\alpha}(w^k - y^k) \\ g \end{bmatrix}, \tag{3.55}$$

which requires pre-computing either matrix inverse [84] or LDL decomposition [134]. If both matrix H and G are time invariant, such pre-computation only needs to be executed once. However, for time varying applications, *e.g.*, those from nonlinear MPC [92], such precomputation needs to be executed every time matrix H or G is updated.

3.4.3 Chambolle & Pock method

Initially developed for computational imaging applications, the *Chambolle & Pock method*, or preconditioned proximal point method solves problem (3.29) as a convex-concave saddle point problem [34]. In particular, the following iterations was introduce in [96] for solving trajectory optimization applications

$$z^{k+1} = \operatorname*{argmin}_{z \in \mathbb{Z}} \tfrac{1}{2} z^\top H z + h^\top z + \tfrac{1}{2\alpha} \left\| z + \alpha G^\top w^k - z^k \right\|_2^2 \tag{3.56a}$$

$$w^{k+1} = w^k + \alpha(G(2z^{k+1} - z^k) - g). \tag{3.56b}$$

Unlike method (3.52), this method does not require matrix H to be positive definite. However, the minimization step in (3.56a) is just as challenging as the one in (3.52a), and can only be solved approximately via Nesterov's method as follows (again, j denotes the inner loop iteration counter)

$$\begin{aligned} z^{j+1} &= \pi_{\mathbb{Z}}[y^j - \tfrac{1}{\lambda}\big(Hy^j + h + G^\top w^k + \tfrac{1}{\alpha}(y^j - z^k)\big)], \\ y^{j+1} &= z^{j+1} + \tfrac{\sqrt{\lambda}-\sqrt{\mu}}{\sqrt{\lambda}+\sqrt{\mu}}(z^{j+1} - z^j), \end{aligned} \tag{3.57}$$

where $0 \prec \mu I \preceq H + \frac{1}{\alpha} I \preceq \lambda I$. Similar to the dual fast gradient method (3.52), the Chambolle & Pock method also requires a double-loop structure.

Lately, a more efficient variant of Chambolle & Pock method was introduced in [35]. Instead of the double-loop iterations in (3.56) and (3.57), this variant uses a single-loop iteration structure. If $H \succeq 0$ in (3.29), this method achieves $O(1/k)$ convergence rate using the following iteration [35, Alg. 1]

$$z^{k+1} = \pi_{\mathbb{Z}}[z^k - \alpha(Hz^k + h + G^\top w^k)] \tag{3.58a}$$

$$w^{k+1} = w^k + \beta(G(2z^{k+1} - z^k) - g). \tag{3.58b}$$

Another variant of the above algorithm was introduced in [21], where (3.58b) is replaced with $w^{k+1} = w^k + \beta(Gz^{k+1} - g)$. However, no convergence rate was proved for this variant. If $H \succ 0$, an improved convergence rate of $O(1/k^2)$ can be achieved using the following iterates [35, Alg. 4]

$$w^{k+1} = w^k + \beta^k(G(z^k + \gamma^k(z^k - z^{k-1})) - g), \tag{3.59a}$$

$$z^{k+1} = \pi_{\mathbb{Z}}[z^k - \tfrac{\alpha^k}{\mu\alpha^k + 1}(Hz^k + h + G^\top w^{k+1})] \tag{3.59b}$$

where step sizes $(\alpha^k, \beta^k, \gamma^k)$ are computed recursively; see [35, Sec. 5.2] for details. The convergence of algorithm (3.58) and (3.59) are both proved using a non-negative running duality gap function [35, Thm. 1, Thm. 4].

Future work Similar to the previous section, this section is also divided into two parts: distributed optimization and trajectory optimization.

Section 3.1 demonstrated how proportioanl-integral control can be used to design distributed optimization algorithms. However, the usefulness of derivative control in distributed optimization algorithms is still unclear. In classical PID control setup, the derivative term focus on the rate of change of the error term, and does not bring a system to its setpoint on its own. hence, one intuitive conjecture is that an additional derivative term in the framework of Section 3.1 will improve the transient convergence behavior of the resulting algorithms. A concrete theoretical or experimental justification of this conjecture can be of great interest in terms of improving algorithm convergence performance.

In addition, Section 3.1 only consider distributed optimization with local set constraints. Such consideration makes its results seamlessly generalizable to trajectory optimization, which is demonstarted in Section 3.2. However, it also makes Section 3.1 miss many recent work on distributed optimization without local constraints. Recently, [162] provided a unified perspective on these work using robust control techniques. However, a theoretical framework that unifies results in [162] and results in Section 3.1 is still an open and intriguing question.

The results in Section 3.2, on the other hand, provides an algorithm that already matches the theoretical optimal complexity bound [136]. In terms of further improving algorithm efficiency, which is the usually the most important performance criterion in application, future work can try to improve the dependency on the parameters of the quadratic objective function, such as the largest and smallest eigenvalues of matrix Q and R. One related direction is to design better algorithms when either matrix has zero eigenvalues.

Another drawback of the results in Section 3.2 is its lack of details on implementation. For example, the bottleneck of implementing the PI projected gradient method in Section 3.2 is computing projections onto state and input constraint set. Although projection onto some simple state or input constraints sets admit close-form solution [8, Chp. 29], projection onto general closed convex sets requires iterative methods. Therefore, developing efficient projection algorithms for state and input sets arising from different application scenarios is the next step to make Section 3.2 more useful in practice.

Further, it is well known that first order optimization method quickly converges to a low or medium accuracy solution, but struggle to converges to high accuracy solution afterwards (also known as the “poor tail convergence"). On the other hand, second order methods based on Newton’s step usually converges quickly from a medium accuracy solution to a high accuracy one, using a much higher per-iteration cost. Hence another future direction of extending Section 3.2 is to implement the PI projected gradient method together together with a second order method. The PI projected gradient method drives the initial guess to the neighborhood of the optimal solution and pass it to the second order method, who converges to a high-accuracy solution using very few iterations.

4 Data-driven approaches

Let us reconsider the trajectory optimization problem (3.1). From Chapter 3, we already know how to efficiently compute a sequence of inputs such that the states of a known dynamics given by

$$x_{t+1} = Ax_t + Bu_t, \tag{4.1}$$

optimally track a reference trajectory.

What if the parameters of the dynamics, *i.e.*, matrices A and B, are unknown?

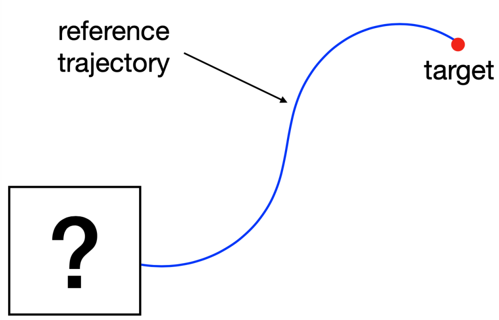


Figure 4.1: Tracking a reference trajectory with unknown dynamics

A recent solution to the above problem is the so-called data-driven predictive control.

Instead of (3.1), this approach solves the following optimization problem [39]

$$\begin{array}{ll} \underset{g,\bar{u}_{[0,L-1]},\bar{x}_{[0,L-1]}}{\text{minimize}} & \sum_{t=0}^{L-1}(\bar{x}_t - r_t)^\top Q_t(\bar{x}_t - r_t) + \sum_{t=0}^{L-1} \bar{u}_t^\top R_t \bar{u}_t \\ \text{subject to} & \begin{bmatrix} u_0 & u_1 & \dots & u_{T-L} \\ \vdots & \vdots & \vdots & \vdots \\ u_{L-1} & u_L & \dots & u_{T-1} \\ x_0 & x_1 & \dots & x_{T-L} \\ \vdots & \vdots & \vdots & \vdots \\ x_{L-1} & x_L & \dots & x_{T-1} \end{bmatrix} g = \begin{bmatrix} \bar{u}_{[0,L-1]} \\ \bar{x}_{[0,L-1]} \end{bmatrix} \\ & \bar{x}_0 = \hat{x}, \\ & \bar{x}_t \in \mathbb{X}_t, \;\; \bar{u}_t \in \mathbb{U}_t, \;\; t \in [0, L-1] \end{array} \tag{4.2}$$

where $(u_{[0,T-1]}, x_{[0,T-1]})$ is a input-state trajectory generated by unknown system (4.1). If system (4.1) is controllable (see Appendix D.1) and the input data trajectory $u_{[0,T-1]}$ is sufficiently persistently exciting (to be defined later), then optimization (4.2) is equivalent to the following optimization

$$\begin{array}{ll} \underset{\bar{u}_{[0,L-1]},\bar{x}_{[0,L-1]}}{\text{minimize}} & \sum_{t=1}^{L-1}(\bar{x}_t - r_t)^\top Q_t(\bar{x}_t - r_t) + \sum_{t=0}^{L-1} \bar{u}_t^\top R_t \bar{u}_t \\ \text{subject to} & \bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t, \;\; \bar{x}_0 = \hat{x}, \\ & \bar{x}_t \in \mathbb{X}_t, \;\; \bar{u}_t \in \mathbb{U}_t, \;\; t \in [0, L-1] \end{array} \tag{4.3}$$

What are the mathematical principles behind this equivalence?

In this chapter, we will show that the above equivalence is due to a classical result in system idetification and data-driven control, named Willems' fundamental lemma. This lemma allows us to “predict" future trajectories of a linear time invariant (LTI) system using its past trajectories. We will introduce a generalized version of Willems' fundamental lemma with weaker assumptions on both the system and the data. Our results will explain the role of controllability and persistent excitationthat in data-driven control more accurately than Willems' original results . We will illustrate the usefulness of our findings via applications in data-driven predictive control and identifying homogeneous multi-agent systems.

We introduce some notations that will later facilitate our discussion. The left kernel of a real matrix M is the set of (real-valued) vectors v such that $v^\top M = 0$. The subspace spanned by the columns of the matrix M is denoted by **col** M, and the Moore-Penrose inverse of M is denoted by $M^\dagger$. When applied to subspaces, we let $+$ and $\times$ denote the sum [78, p.2] and Cartesian product operation [78, p.370], respectively. Given a signal $f : \mathbb{N} \to \mathbb{R}^q$ and $i, j \in \mathbb{N}$ with $i \leqslant j$, we denote $f_{[i,j]} = \begin{bmatrix} f_i^\top & f_{i+1}^\top & \cdots & f_j^\top \end{bmatrix}^\top$, and the Hankel matrix of depth d $(k \leqslant j - i + 1)$ associated with $f_{[i,j]}$ as

$$H_d(f_{[i,j]}) = \begin{bmatrix} f_i & f_{i+1} & \cdots & f_{j-d+1} \\ f_{i+1} & f_{i+2} & \cdots & f_{j-d+2} \\ \vdots & \vdots & \vdots & \vdots \\ f_{i+d-1} & f_{i+d} & \cdots & f_j \end{bmatrix}.$$

Given matrix $A \in \mathbb{R}^{n\times n}$ and a monic polynomial of degree δ, *i.e.*, $q(s) = s^\delta + \sum_{i=0}^{\delta-1} \alpha_0 s^i$ where $\alpha_0, \alpha_1, \ldots, \alpha_{\delta-1} \in \mathbb{R}$, we say monic polynomial$q(s)$ annihilates matrix A if

$$q(A) = A^\delta + \sum_{i=0}^{\delta-1} \alpha_i A^i = 0_{n\times n}.$$

Further, for each matrix $A \in \mathbb{R}^{n\times n}$, there exists an unique monic polynomial of minimal degree that annihilates matrix A [78, Def. 3.3.2], which is called the *minimal polynomial* of matrix A.

4.1 Generalized Willems' fundamental lemma

Throughout we consider consider the following LTI system

$$x_{t+1} = Ax_t + Bu_t, \tag{4.4a}$$

$$y_t = Cx_t + Du_t, \tag{4.4b}$$

where $u_t \in \mathbb{R}^m, x_t \in \mathbb{R}^n, y_t \in \mathbb{R}^p$ denote the input, state and output of the system at discrete time $t \in \mathbb{N}$, respectively. Notice that the states are the outputs for the case where $C = I$ and D.

Throughout we let

$$(u^i_{[0,T^i-1]}, x^i_{[0,T^i-1]}, y^i_{[0,T^i-1]})$$

denote a length-T^i ($T^i \in \mathbb{N}_+$) input-state-output trajectory generated by system (4.4) for all $i = 1, 2, \ldots, \tau$, where $\tau \in \mathbb{N}_+$ is the total number of trajectories. We let

$$\{u^i_{[0,T^i-1]}\}_{i=1}^{\tau}, \;\; \{x^i_{[0,T^i-1]}\}_{i=1}^{\tau}, \;\; \{y^i_{[0,T^i-1]}\}_{i=1}^{\tau}, \tag{4.5}$$

denote the set of input, state, and output trajectories, respectively. We will use the following subspaces

$$\begin{aligned} \mathcal{R} &= \operatorname{im}\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}, \\ \mathcal{O} &= \ker\begin{bmatrix} C^\top & (CA)^\top & \cdots & (CA^{n-1})^\top \end{bmatrix}^\top, \\ \mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau] &= \operatorname{im}\begin{bmatrix} X_0 & AX_0 & \cdots & A^{n-1}X_0 \end{bmatrix}, \end{aligned} \tag{4.6}$$

where $X_0 = \begin{bmatrix} x_0^1 & x_0^2 & \cdots & x_0^\tau \end{bmatrix}$.In particular, $\mathcal{R}$ is known as the controllable subspace, $\mathcal{O}$ is known as the unobservable subspace, $\mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau]$ is the controllable subspace with B replaced by X_0. We say system (4.4) is controllable if $\mathcal{R} = \mathbb{R}^n$. One can verify that $\mathcal{R}$, $\mathcal{O}$ and $\mathcal{K}[x_0^1, x_0^2, \cdots, x_0^\tau]$ are all invariant subspace of matrix A.

We will also use the following definitions that streamline the subsequent analysis.

Definition 4.1. *We say a length-L input-output trajectory $(\overline{u}_{[0,L-1]}, \overline{y}_{[0,L-1]})$ with $L \in \mathbb{N}_+$ is* parameterizable *by $\{u_{[0,T^i-1]}, y_{[0,T^i-1]}\}_{i=1}^{\tau}$ if there exists $g \in \mathbb{R}^{\sum_{i=1}^{\tau}(T^i-L+1)}$ such that*

$$\begin{bmatrix} \overline{u}_{[0,L-1]} \\ \overline{y}_{[0,L-1]} \end{bmatrix} = \begin{bmatrix} H_L(u^1_{[0,T^1-1]}) & \cdots & H_L(u^\tau_{[0,T^\tau-1]}) \\ H_L(y^1_{[0,T^1-1]}) & \cdots & H_L(y^\tau_{[0,T^\tau-1]}) \end{bmatrix} g. \tag{4.7}$$

As an example, if $\tau = 2$, $L = 2$, $T^1 = 3, T^2 = 4$, then equation (4.7) becomes the following

$$\left[\begin{array}{c} \bar{u}_0 \\ \bar{u}_1 \\ \hdashline \bar{y}_0 \\ \bar{y}_1 \end{array}\right] = \left[\begin{array}{cc:ccc} u_0^1 & u_1^1 & u_0^2 & u_1^2 & u_2^2 \\ u_1^1 & u_2^1 & u_1^2 & u_2^2 & u_3^2 \\ \hdashline y_0^1 & y_1^1 & y_0^2 & y_1^2 & y_2^2 \\ y_1^1 & y_2^1 & y_1^2 & y_2^2 & y_3^2 \end{array}\right] g.$$

Definition 4.2 (Collective persistent excitation [166])**.** *We say* $\{u_{[0,T^i-1]}\}_{i=1}^{\tau}$ *is* collectively persistently exciting *of order* $d \in \mathbb{N}_+$ *if* $d \leqslant T^i$ *for all* $i = 1, 2, \ldots, \tau$ *and the mosaic-Hankel matrix, defined as*

$$\left[H_d(u^1_{[0,T^1-1]}) \quad \cdots \quad H_d(u^\tau_{[0,T^\tau-1]})\right], \tag{4.8}$$

has full row rank.

If $\tau = 1$, then Definition 4.2 reduces to the traditional notion of persistency of excitation.

The Willems' fundamental lemma asserts that: if system (4.4) is controllable and $\{u^i_{[0,T^i-1]}\}_{i=1}^{\tau}$ is collectively persistently exciting of order $n+L$, then $(\bar{u}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ is parameterizable by $\{u_{[0,T^i-1]}, y_{[0,T^i-1]}\}_{i=1}^{\tau}$ if and only if it is an input-output trajectory of system (4.4) [182, 166]. As our main contribution, the following theorem shows that not only this lemma can be extended to an arbitrary LTI system, but also the required order of collective persistent excitation can be reduced from $n + L$ to $\delta_{\min} + L$, where $\delta_{\min}$ is the degree of the minimal polynomial of matrix A.

With the above definitions, we are now ready to introduce the main result of this chapter, a generalized version of Willems' fundamental lemma in [182].

Theorem 4.3. *Let* $\delta_{\min}$ *be the degree of the minimal polynomial of matrix* A*, and* $\delta \in \mathbb{N}_+$ *with* $\delta \geqslant \delta_{\min}$*. Let* $\{u^i_{[0,T^i-1]}, x^i_{[0,T^i-1]}, y^i_{[0,T^i-1]}\}_{i=1}^{\tau}$ *be the set of input-state-output trajectories generated by system* (4.4)*. If* $\{u_{[0,T^i-1]}\}_{i=1}^{\tau}$ *is collectively persistently exciting of order* $\delta + L$*,* *then*

$$\operatorname{im}\begin{bmatrix} H_1(x^1_{[0,T^1-L]}) & \cdots & H_1(x^\tau_{[0,T^\tau-L]}) \\ H_L(u^1_{[0,T^1-1]}) & \cdots & H_L(u^\tau_{[0,T^\tau-1]}) \end{bmatrix} = (\mathcal{R}+\mathcal{K}[x^1_0, x^2_0, \ldots, x^\tau_0]) \times \mathbb{R}^{mL}. \tag{4.9}$$

Further, $(\bar{u}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ *is parameterizable by* $\{u_{[0,T^i-1]}, y_{[0,T^i-1]}\}_{i=1}^{\tau}$ *if and only if there exists a state trajectory* $\bar{x}_{[0,L-1]}$ *with*

$$\bar{x}_0 \in \mathcal{R}+\mathcal{O}+\mathcal{K}[x^1_0, x^2_0, \ldots, x^\tau_0], \tag{4.10}$$

such that $(\bar{u}_{[0,L-1]}, \bar{x}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ *is an input-state-output trajectory of system* (4.4).

Proof. We start by proving the first statement using a double inclusion argument. It is trivial to show that the left hand side of (4.9) is included in its right hand side. To show the other direction, we show that the left kernel of matrix

$$\begin{bmatrix} H_1(x^1_{[0,T^1-L]}) & \cdots & H_1(x^\tau_{[0,T^\tau-L]}) \\ H_L(u^1_{[0,T^1-1]}) & \cdots & H_L(u^\tau_{[0,T^\tau-L]}) \end{bmatrix} \tag{4.11}$$

is orthogonal to $(\mathcal{R}+\mathcal{K}[x^1_0, x^2_0, \ldots, x^\tau_0]) \times \mathbb{R}^{mL}$, To this end, let

$$v^\top = \begin{bmatrix} \xi^\top & \eta_1^\top & \eta_2^\top & \cdots \eta_L^\top \end{bmatrix} \tag{4.12}$$

be an arbitrary row vector in the left kernel of matrix (4.11), where $\xi \in \mathbb{R}^n, \eta_1, \eta_2, \ldots, \eta_L \in \mathbb{R}^m$. Since $\delta \geqslant \delta_{\min}$, using [78, Def. 3.3.2] we know there exists $\alpha_{0k}, \alpha_{1k}, \ldots, \alpha_{\delta-1,k} \in \mathbb{R}$ such that

$$A^k + \textstyle\sum_{j=0}^{\delta-1} \alpha_{jk} A^j = 0_{n\times n}, \quad \forall k = \delta, \delta+1, \ldots \tag{4.13}$$

The above equation implies that $A^k B = -\sum_{j=0}^{\delta-1} \alpha_{jk} A^j B$ and $A^k x^i_0 = -\sum_{j=0}^{\delta-1} \alpha_{jk} A^j x^i_0$ for all $k = \delta, \ldots, n-1$ and $i = 1, \ldots, \tau$. Therefore in order to show the left kernel of matrix (4.11) is orthogonal to $(\mathcal{R}+\mathcal{K}[x_0^1, x_0^2, \dots, x_0^\tau]) \times \mathbb{R}^{mL}$, it suffices to show the following

$$\eta_1^\top = \eta_2^\top = \cdots = \eta_L^\top = 0_m^\top, \tag{4.14a}$$

$$\xi^\top B = \xi^\top AB = \cdots = \xi^\top A^{\delta-1}B = 0_m^\top, \tag{4.14b}$$

$$\xi^\top x_0^i = \xi^\top Ax_0^i = \cdots = \xi^\top A^{\delta-1}x_0^i = 0. \ \forall i = 1, \dots, \tau. \tag{4.14c}$$

In order to prove (4.14a), we let $w_0, w_1, \dots, w_\delta \in \mathbb{R}^{n+m(\delta+L)}$ be such that $w_0 = \begin{bmatrix} v^\top & 0_{m\delta}^\top \end{bmatrix}^\top$ and w_j equals

$$\begin{bmatrix} \xi^\top A^j & \xi^\top A^{j-1}B & \cdots \xi^\top B & \eta_1^\top & \cdots & \eta_L^\top & 0_{m(\delta-j)}^\top \end{bmatrix}^\top,$$

for $j = 1, \dots, \delta$. Since $v^\top$ is in the left kernel of matrix (4.11), using (4.4) one can verify that $w_0^\top, w_1^\top, \dots, w_\delta^\top$ are in the left kernel of

$$\begin{bmatrix} H_1(x_{[0,T^1-\delta-L]}^1) & \cdots & H_1(x_{[0,T^\tau-\delta-L]}^\tau) \\ H_{\delta+L}(u_{[0,T^1-1]}^1) & \cdots & H_{\delta+L}(u_{[0,T^\tau-1]}^\tau) \end{bmatrix}. \tag{4.15}$$

Let $\alpha_{\delta\delta} = 1$ and $k = \delta$ in (4.13), we have $0_{n\times n} = \sum_{j=0}^{\delta} \alpha_{j\delta} A^j$. Hence

$$\textstyle\sum_{j=0}^{\delta} \alpha_{j\delta} w_j^\top = \begin{bmatrix} \sum_{j=0}^{\delta} \alpha_{j\delta} \xi^\top A^j & r^\top \end{bmatrix} = \begin{bmatrix} 0_n & r^\top \end{bmatrix}, \tag{4.16}$$

for some vector $r \in \mathbb{R}^{m(\delta+L)}$. Since row vectors $w_0^\top, w_1^\top, \dots, w_\delta^\top$ are in the left kernel of matrix (4.15), equation (4.16) implies that $r^\top$ is in the left kernel of matrix

$$\begin{bmatrix} H_{\delta+L}(u_{[0,T^i-1]}^1) & \cdots & H_{\delta+L}(u_{[0,T^i-1]}^\tau) \end{bmatrix}. \tag{4.17}$$

Since $\{u_{[0,T^i-1]}^i\}_{i=1}^\tau$ is collectively persistently exciting of order $\delta + L$, matrix (4.17) has full row rank. Therefore

$$r = 0_{m(\delta+L)}. \tag{4.18}$$

Observe that the last m entries of r are given by $\alpha_{\delta\delta}\eta_L = \eta_L$, hence equation (4.18) implies that $\eta_L = 0_m$. Then the last $2m$ entries of r are given by $\begin{bmatrix}\alpha_{\delta\delta}\eta_{L-1}^\top + \alpha_{(\delta-1)\delta}\eta_L^\top & \alpha_{\delta\delta}\eta_L^\top\end{bmatrix}^\top$. Since $\eta_L = 0_m$ and $\alpha_{\delta\delta} = 1$, equation (4.18) also implies that $\eta_{L-1} = 0_m$. By repeating similar induction we can prove that (4.14a) holds.

Next, since (4.14a) holds, the first $m\delta$ entries in r are

$$\begin{bmatrix}\sum_{j=1}^{\delta}\alpha_{j\delta}\xi^\top A^{j-1}B & \sum_{j=2}^{\delta}\alpha_{j\delta}\xi^\top A^{j-2}B & \cdots & \alpha_{\delta\delta}\xi^\top B\end{bmatrix}^\top.$$

By combining this with (4.18) we get that

$$0_m^\top = \textstyle\sum_{j=k}^{\delta}\alpha_{j\delta}\xi^\top A^{j-k}B, \;\; \forall k = 1, \dots, \delta. \tag{4.19}$$

Since $\alpha_{\delta\delta} = 1$, considering $k = \delta$ in (4.19) implies that $\xi^\top B = 0_m$. Substitute this back into (4.19) and considering $k = \delta - 1$ implies that $\xi^\top AB = 0_m$. By repeating a similar induction we can prove that (4.14b) holds.

Further, by using (4.13) and (4.14b) we can show that $\xi^\top A^k B = 0_m$ for all $k \geqslant 1$. Combining this together with the fact that row vector (4.12) is in the left kernel of matrix (4.11) and that (4.14a) also holds, it follows that,

$$\begin{aligned}0 =&\xi^\top x_k^i = \xi^\top\big(A^k x_0^i + \textstyle\sum_{j=0}^{k-1}A^{k-j-1}Bu_j^i\big) = \xi^\top A^k x_0^i,\\ &\forall k = 0, \dots, T^i - L, \;\; i = 1, \dots, \tau.\end{aligned}$$

Since $T^i \geqslant \delta + L$ for $i = 1, \dots, \tau$ by assumption, we conclude that (4.14c) holds.

We now prove the second statement. Given the input, state, and output trajectories in (4.5), suppose (4.7) holds. Let

$$\begin{aligned}\bar{x}_0 =& \begin{bmatrix}H_1(x^1_{[0,T^1-L]}) & \cdots & H_1(x^\tau_{[0,T^\tau-L]})\end{bmatrix} g,\\ \bar{x}_{t+1} =& A\bar{x}_t + B\bar{u}_t, \;\; 0 \leqslant t \leqslant L-2.\end{aligned} \tag{4.20}$$

Then from (4.9) we know $\bar{x}_0$ satisfy (4.10), and one can verify that $(\bar{u}_{[0,L-1]}, \bar{x}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ is indeed an input-state-output trajectory of system (4.4).

Conversely, let $(\bar{u}_{[0,L-1]}, \bar{x}_{[0,L-1]}, \bar{y}_{[0,L-1]})$ be an input-state-output trajectory of system (4.4) with $\bar{x}_0 \in \mathcal{R} + \mathcal{O} + \mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau]$. Then there exists

$$\bar{x}_0^a \in \mathcal{R} + \mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau], \;\; \bar{x}_0^b \in \mathcal{O}, \tag{4.21}$$

such that $\bar{x}_0 = \bar{x}_0^a + \bar{x}_0^b$ and

$$\begin{bmatrix} \bar{u}_{[0,L-1]} \\ \bar{y}_{[0,L-1]} \end{bmatrix} = \begin{bmatrix} 0 & I \\ O_L & T_L \end{bmatrix} \begin{bmatrix} \bar{x}_0^a + \bar{x}_0^b \\ \bar{u}_{[0,L-1]} \end{bmatrix} \tag{4.22}$$

where

$$\begin{aligned} T_L &= \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{L-2}B & CA^{L-3}B & CA^{L-4}B & \cdots & D \end{bmatrix}, \\ O_L &= \begin{bmatrix} C^\top & (CA)^\top & (CA^2)^\top & \cdots & (CA^{L-1})^\top \end{bmatrix}^\top . \end{aligned} \tag{4.23}$$

Further, using the Cayley-Hamilton theorem one can show that $\mathcal{O} \subset \ker O_L$ for any $L \in \mathbb{N}_+$. Hence (4.21) implies

$$O_L x_0^b = 0_{Lp}. \tag{4.24}$$

Since $\bar{x}_0^a \in \mathcal{R}+\mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau]$, the first statement implies that there exists $g \in \mathbb{R}^{\sum_{i=1}^{\tau}(T^i-L+1)}$ such that

$$\begin{bmatrix} \bar{x}_0^a \\ \bar{u}_{[0,L-1]} \end{bmatrix} = \begin{bmatrix} H_1(x^1_{[0,T^1-L]}) & \cdots & H_1(x^\tau_{[0,T^\tau-L]}) \\ H_L(u^1_{[0,T^1-1]}) & \cdots & H_L(u^\tau_{[0,T^\tau-1]}) \end{bmatrix} g. \tag{4.25}$$

Notice that
$$\begin{bmatrix} 0 & I \\ O_L & T_L \end{bmatrix} \begin{bmatrix} H_1(x^i_{[0,T^i-L]}) \\ H_L(u^i_{[0,T^i-1]}) \end{bmatrix} = \begin{bmatrix} H_L(u^i_{[0,T^i-1]}) \\ H_L(y^i_{[0,T^i-1]}) \end{bmatrix}. \tag{4.26}$$
for $i = 1, \dots, \tau$. Substituting (4.24), (4.25) and (4.26) into (4.22) gives (4.7), thus completing the proof.

□

Remark 4.1. *The equality in* (4.9) *generalizes [123, Lem. 2] by proving stronger results using weaker assumptions. Particularly, the assumption of $n + L$ order of persistently excitation in [123, Lem. 2] is reduced to $\delta + L$ with $\delta \geqslant \delta_{\min}$, and the controllable subspace $\mathcal{R}$ used in [123, Lem. 2] is extended to its superset $\mathcal{R} + \mathcal{K}[x_0^1, x_0^2, \dots, x_0^\tau]$.*

Remark 4.2. *Theorem 4.3 generalizes [166, Thm. 2] by proving the same results using weaker assumptions. Particularly, to ensure all input-output trajectories generated by system* (4.4) *are parameterizable by a finite number of them, [166, Thm. 2] assumes $\mathcal{R} = \mathbb{R}^n$. In comparison, using Theorem 4.3 one only need to assume that $\mathcal{R} + \mathcal{O} + \mathcal{K}[x_0^1, x_0^2, \dots, x_0^\tau] = \mathbb{R}^n$ to ensure the same results.*

The first statement in Theorem 4.3 leads to a new result in system identification, summarized by the following corollary.

Corollary 4.1. *Let $\{u^i_{[0,T^i-1]}, x^i_{[0,T^i]}\}_{i=1}^\tau$ be input-state trajectories generated by system* (4.4a)*, and input sequences $\{u^i_{[0,T^i-1]}\}_{i=1}^\tau$ are collectively persistently exciting of order $\delta + 1$ with $\delta \geqslant \delta_{\min}$, where $\delta_{\min}$ is the degree of the minimal polynomial of A. Define $\begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} := XS^\dagger$, where,*

$$X = \begin{bmatrix} H_1(x^1_{[1,T^1]}) & \cdots & H_1(x^\tau_{[1,T^\tau]}) \end{bmatrix}, \tag{4.27a}$$
$$S = \begin{bmatrix} H_1(x^1_{[0,T^1-1]}) & \cdots & H_1(x^\tau_{[0,T^\tau-1]}) \\ H_1(u^1_{[0,T^1-1]}) & \cdots & H_1(u^\tau_{[0,T^\tau-1]}) \end{bmatrix}. \tag{4.27b}$$

Then $B = \hat{B}$ and $Az = \hat{A}z$ for any $z \in \mathcal{R} + \mathcal{K}[x_0^1, x_0^2, \dots, x_0^\tau]$.

Proof. The first statement of Theorem 4.3 implies that, for any $z \in \mathcal{R} + \mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau]$ and $v \in \mathbb{R}^m$, there exists $g \in \mathbb{R}^{\sum_{i=1}^{\tau} T^i}$ such that $\begin{bmatrix} z \\ v \end{bmatrix} = Sg$. Using the definition of matrix S and X we can show

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} A & B \end{bmatrix} Sg = Xg.$$

Further, notice that

$$\begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} = XS^\dagger Sg = Xg = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix}.$$

This completes the proof by considering the following two cases where: 1) $z = 0_n$ and v ranges over $\mathbb{R}^m$, and 2) $v = 0_m$ and z ranges over $\mathcal{R} + \mathcal{K}[x_0^1, x_0^2, \ldots, x_0^\tau]$. □

Remark 4.3. *If $\delta = n$ and system* (4.4) *is controllable, then Corollary 4.1 reduces to Theorem 1 in [42].*

Another implication of Theorem 4.3 is that the order of persistent excitation required by trajectory parameterization only depends on the degree of the minimal polynomial of matrix A in (4.4), instead of its dimension. In general, it is difficult to establish a bound of the degree of the minimal polynomial of a matrix tighter than its dimension. However, the following corollary shows an exception example; its usefulness will be illustrated later in Section 4.2.2.

Corollary 4.2. *If there exists $\bar{A} \in \mathbb{R}^{\frac{n}{N} \times \frac{n}{N}}$ such that $A = I_N \otimes \bar{A}$, then Theorem 4.3 holds with $\delta = \frac{n}{N}$.*

Proof. The results directly follow from Theorem 4.3 and the fact the minimal polynomial of A is the same as the one of $\bar{A}$, which has degree at most $\frac{n}{N}$. □

4.2 Applications

In this section, we provide two examples that illustrate distinct implications of Theorem 4.3.

4.2.1 Online data-driven predictive control

Model predictive control (MPC) provides an effective strategy for systems with physical and operational constraints [120, 119]. In particular, consider system (4.4a). At each sampling time t, MPC solves the following optimization to obtain the input u_t

$$\begin{aligned} \underset{\substack{\bar{u}_{[t,t+L-1]} \\ \bar{x}_{[t,t+L-1]}}}{\text{minimize}} \quad & \textstyle\sum_{k=t}^{t+L-1} \left(\|\bar{x}_k - r_k\|_Q^2 + \|\bar{u}_k\|_R^2 \right) \\ \text{subject to} \quad & \bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k, \;\; \bar{x}_t = \hat{x}_t, \\ & \bar{x}_k \in \mathbb{X}, \;\; \bar{u}_k \in \mathbb{U}, \;\; k = t, \ldots, t+L-1, \end{aligned} \tag{4.28}$$

where $\hat{x}_t \in \mathbb{R}^n$ is the current state and $L \in \mathbb{N}_+$ is the planning horizon. Closed convex sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ describe feasible states and inputs, respectively. Symmetric positive semi-definite weighting matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$, together with reference state trajectory $r_{[t,t+L-1]}$, define the quadratic tracking cost function.

Recently, [39, 4] proposed data-driven predictive control (DPC) that replaces optimization (4.28) with

$$\begin{aligned} \underset{\substack{g, \bar{u}_{[t,t+L-1]} \\ \bar{x}_{[t,t+L-1]}}}{\text{minimize}} \quad & \textstyle\sum_{k=t}^{t+L-1} \left(\|\bar{x}_k - r_k\|_Q^2 + \|\bar{u}_k\|_R^2 \right) \\ \text{subject to} \quad & \begin{bmatrix} \bar{u}_{[t,t+L-1]} \\ \bar{x}_{[t,t+L-1]} \end{bmatrix} = \begin{bmatrix} H_L(u_{[0,T-1]}) \\ H_L(x_{[0,T-1]}) \end{bmatrix} g \\ & \bar{x}_t = \hat{x}_t, \\ & \bar{x}_k \in \mathbb{X}, \;\; \bar{u}_k \in \mathbb{U}, \;\; k = t, \ldots, t+L-1, \end{aligned} \tag{4.29}$$

where $(u_{[0,T-1]}, x_{[0,T-1]})$ is an input-state trajectory of system (4.4a) and generated *offline*. If $u_{[0,T-1]}$ is persistently exciting of order $n + L$ and system (4.4a) is controllable, Willems' fundamental lemma guarantees that optimization (4.29) is equivalent to the one in (4.28).

However, testing the controllability of system (4.4a) using its input-state data $(u_{[0,T-1]}, x_{[0,T-1]})$ is expensive: its computation time scales cubically with T [167, 123]. Further, if system (4.4a) is uncontrollable, then Willems' fundamental lemma provides no guarantee on the equivalence between optimization (4.29) and (4.28).

On the other hand, Theorem 4.3 shows that the assumption of (A, B) being controllable can be replaced by $\hat{x}_t \in \mathcal{R} + \mathcal{K}[x_0]$. In particular, if there exists an input sequence $\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_{k-1} \in \mathbb{R}^m$ such that

$$\hat{x}_t = A^k x_0 + \textstyle\sum_{j=0}^{k-1} A^{k-j-1} B \tilde{u}_j. \tag{4.30}$$

Using the Cayley-Hamilton theorem, one can verify that $\hat{x}_t \in \mathcal{R} + \mathcal{K}[x_0]$ for any $k \in \mathbb{N}_+$. With this observation, we propose the following *online DPC* algorithm:

1. At time $t = 1, 2, \ldots, T-1$, generate input-state data $(u_{[0,T-1]}, x_{[0,T-1]})$ such that $u_{[0,T-1]}$ is persistently exciting of order $n + L$,
2. At time $t = T, T+1, \ldots$, compute the input $\overline{u}_t$ by solving optimization (4.29) given the current state $\hat{x}_t$ and input-state data $(u_{[0,T-1]}, x_{[0,T-1]})$.

Notice that this online DPC algorithm ensures that

$$\hat{x}_t = A^t x_0 + \textstyle\sum_{i=0}^{T-1} A^{T-i-2} B u_i + \sum_{j=T}^{t} A^{t-j-1} \overline{u}_j, \tag{4.31}$$

for all $t \geqslant T$, *i.e.*, condition (4.30) is satisfied with input sequence $u_0, u_1, \ldots, u_{T-1}, \overline{u}_T, \overline{u}_{T+1}, \ldots, \overline{u}_t$. From Theorem 4.3, we know optimization (4.29) solved in the above online DPC algorithm is equivalent to (4.28), regardless of the controllability of system (4.4a).

Consider the system in (4.4a), for example, with

$$A = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0 & 0.9 \end{bmatrix}, \quad B = \begin{bmatrix} 0.125 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}. \tag{4.32}$$

Let $[x]_i$ denote the i-th coordinate of the state. One can verify that $[x]_3$ and $[x]_4$ are uncontrollable modes, *i.e.*, they evolve independently from the inputs. We choose $L = 5$

and generate online input-state trajectories $(u_{[0,T-1]}, x_{[0,T-1]})$, where u_t is sampled uniformly from $[-0.04, 0.04]$ for all $t = 0, \ldots, T-1$, until $u_{[0,T-1]}$ is persistently exciting of order 9. At time $t = T, T+1, \ldots$, given the current state $\hat{x}_t$, we obtain the input $\overline{u}_t$ by solving (4.29) where $Q = I_4$, $R = 0.001$, $\mathbb{X} = \mathbb{R}^4$ and $\mathbb{U} = \{u \in \mathbb{R} | -0.5 \leqslant u \leqslant 0.5\}$. Fig. 4.2 shows that, although the system contains uncontrollable modes, online DPC still ensures that the controllable modes, *e.g.*, $[x]_1$, track the reference trajectory as desired.

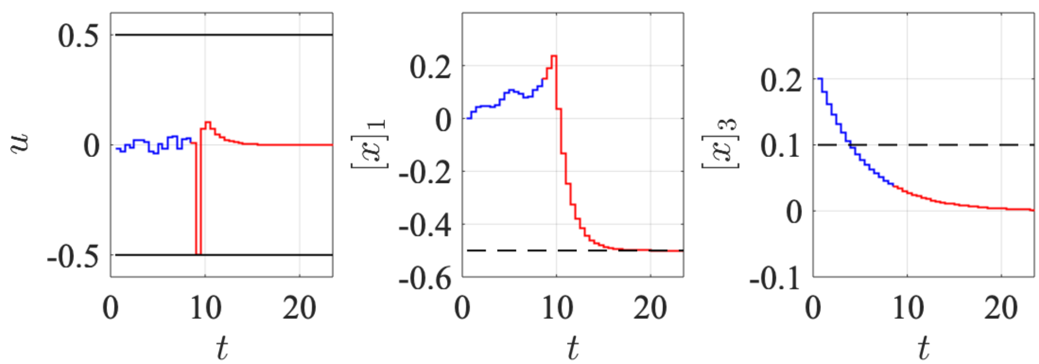


Figure 4.2: Online DPC for (A, B) in (4.32). The blue and red curves denote data and online DPC trajectories, respectively; the solid and dashed lines denote input bounds and state reference values, respectively.

4.2.2 Identification of homogeneous multi-agent systems

Consider a network of N agents with the same LTI dynamics [179]. Further, agent i can measure the state of agent j in a local coordinate system if (i, j) is an edge of a directed graph $\mathcal{G}$, which is composed of N nodes and M edges. The dynamics of this multi-agent system is given by (4.4) with

$$A = I_N \otimes \bar{A}, B = I_N \otimes \bar{B}, C = E \otimes I_{\bar{n}}, D = 0_{M\bar{n} \times N\bar{m}}, \tag{4.33}$$

where $\bar{A} \in \mathbb{R}^{\bar{n} \times \bar{n}}$ and $\bar{B} \in \mathbb{R}^{\bar{n} \times \bar{m}}$ describe the dynamics of an individual agent. Each row of matrix $E \in \mathbb{R}^{M \times N}$ is indexed by an directed edge, *i.e.*, an edge with an head and a tail, in graph $\mathcal{G}$: the i-th entry in each row is "1" if node i is the head of the corresponding edge, "-1" if it is the tail, and "0" otherwise. We assume that $\bar{B}$ is a non-zero matrix and $(\bar{A}, \bar{B})$ is controllable.

If at least one non-zero entry in matrix E is known, then system matrices in (4.33) can be computed using the following *Markov parameters* [168, Sec. 3.4.4]

$$\begin{aligned} M_k =& CA^{k-1}B + D \\ =& (E \otimes I_{\bar{n}})(I_N \otimes \bar{A})^{k-1}(I_N \otimes \bar{B}) \\ =& E \otimes (\bar{A}^{k-1}\bar{B}), \quad \forall k = 1, 2, \ldots, \bar{n}+1. \end{aligned} \tag{4.34}$$

In particular, let $(M_k)_{ij}$ denote the ij-th $\bar{n} \times \bar{m}$ block of M_k. If we know $E_{ij} = 1$ (the case of "-1" is similar), then (4.34) implies $(M_k)_{ij} = \bar{A}^{k-1}\bar{B}$. For example, if $E = \begin{bmatrix} 1 & -1 \end{bmatrix}$, then (4.34) says $M_k = \begin{bmatrix} \bar{A}^{k-1}\bar{B} & -\bar{A}^{k-1}\bar{B} \end{bmatrix}$. Hence given the Markov parameters (4.34) and that $E_{ij} = 1$, we know E_{kl} is "1" if $(M_1)_{kl} = (M_1)_{ij}$, "-1" if $(M_1)_{kl} = -(M_1)_{ij}$, and "0" otherwise. Further, $\bar{B} = (M_1)_{ij}$ and $\bar{A}$ is the unique solution to the following linear equations[1]

$$\bar{A}\begin{bmatrix} (M_1)_{ij} & \cdots & (M_{\bar{n}})_{ij} \end{bmatrix} = \begin{bmatrix} (M_2)_{ij} & \cdots & (M_{\bar{n}+1})_{ij} \end{bmatrix}. \tag{4.35}$$

Therefore, given at least one non-zero entry in matrix E, in order to compute the system matrices in (4.33), it suffices to know the Markov parameters (4.34).

To compute the Markov parameters from measured input-output trajectories, we use Corollary 4.2 together with a data-driven simulation procedure [117, 116] described as follows. Let $\{u^i_{[0,T^i-1]}, y^i_{[0,T^i-1]}\}_{i=1}^{\tau}$ be input-output trajectories of the system described by (4.33), such that inputs $\{u^i_{[0,T^i-1]}\}_{i=1}^{\tau}$ are collectively persistently exciting of order $(N+1)\bar{n}+1$. Let $n = N\bar{n}, m = N\bar{m}, p = M\bar{n}$. Since $(\bar{A}, \bar{B})$ in (4.33) is controllable, one can verify that $\mathcal{R} + \mathcal{K}[x^1_0, x^2_0, \ldots, x^{\tau}_0] = \mathbb{R}^n$, regardless of the values of $x^1_0, x^2_0, \ldots, x^{\tau}_0$. Using Corollary 4.2 we

---

[1]Since $(\bar{A}, \bar{B})$ is controllable, matrix $\begin{bmatrix} (M_1)_{ij} & \cdots & (M_{\bar{n}-1})_{ij} \end{bmatrix} = \begin{bmatrix} \bar{B} & \cdots & \bar{A}^{\bar{n}}\bar{B} \end{bmatrix}$ has full column rank and (4.35) has a unique solution.

know that there exists a matrix $G_k \in \mathbb{R}^{(\sum_{i=1}^{\tau}(T^i-n))\times m}$ such that,

$$\begin{aligned}
&\begin{bmatrix} H_{n+1}(u^1_{[0,T^1-1]}) & \cdots & H_{n+1}(u^\tau_{[0,T^\tau-1]}) \\ H_{n+1}(y^1_{[0,T^1-1]}) & \cdots & H_{n+1}(y^\tau_{[0,T^\tau-1]}) \end{bmatrix} G_k \\
&= \begin{bmatrix} 0^\top_{m(n-k)\times m} & I_m & 0^\top_{(pn+km-kp)\times m} & M_0^\top & \cdots & M_k^\top \end{bmatrix}^\top
\end{aligned} \tag{4.36}$$

for all $k = 0, 1, \ldots, \bar{n}+1$, where $M_0 = 0_{p\times m}$ and M_k is given by (4.34); also see [116, Sec. 4.5] for details. Next, given $M_0, \ldots, M_{k-1}$, we can compute M_k by first solving the first $m(n+1)+pn$ equations in (4.36) for matrix G_k, then substituting the solution into the last p equations in (4.36). By repeating this process for $k = 1, \ldots, \bar{n}+1$ we obtain Markov parameters (4.34). Using Kalman decomposition one can verify that Markov parameters of system (4.33) are the same as those of a reduced order controllable and observable system with state dimension less than n. Hence matrix M_k obtained this way is unique; see [116, Prop. 1].

In numerical simulations, we consider the homogeneous multi-agent system used in [179, Example 3], discretized with a sampling time of 0.1s such that the system dynamics is given by (4.33) where

$$\bar{A} = \begin{bmatrix} 0.9964 & 0.0026 & -0.0004 & -0.0460 \\ 0.0045 & 0.9037 & -0.0188 & -0.3834 \\ 0.0098 & 0.0339 & 0.9383 & 0.1302 \\ 0.0005 & 0.0017 & 0.0968 & 1.0067 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0.0445 & 0.0167 \\ 0.3407 & -0.7249 \\ -0.5278 & 0.4214 \\ -0.0268 & 0.0215 \end{bmatrix}.$$

In addition, we let $E = \begin{bmatrix} \mathbf{1}_{N-1} & -I_{N-1} \end{bmatrix}$, where $\mathbf{1}_{N-1} \in \mathbb{R}^{N-1}$ is the vector of all 1's.

We compare Corollary 4.2 against the results in [166, Thm. 2] in terms of the least amount of input-output data needed to compute matrices in (4.33). In particular, since matrix $\bar{A}$ has spectrum radius 1.03, we use use input-output trajectories $\{u^i_{[0,T-1]}, y^i_{[0,T-1]}\}_{i=1}^{\tau}$ with relatively short length $T = 120$ to avoid numerical instability [166]. The entries in $\{u^i_{[0,T-1]}\}_{i=1}^{\tau}$ are sampled uniformly from $[-0.1, 0.1]$. Using Corollary 4.2, the data-driven simulation procedure requires $\{u^i_{[0,T-1]}\}_{i=1}^{\tau}$ to be collectively persistently exciting of order $(N+1)\bar{n}+1$. In other words, matrix (4.8) with $d=(N+1)\bar{n}+1$ has full row rank, hence it must have at least as many columns as rows, *i.e.*,

$$\tau \geqslant \tfrac{((N+1)\bar{n}+1)N\bar{m}}{T-(N+1)\bar{n}} = \tfrac{8N^2+10N}{116-4N}.$$

In comparison, if we use [166, Thm. 2] instead of Corollary 4.2, we need $\{u^i_{[0,T-1]}\}_{i=1}^{\tau}$ to be collectively persistently exciting of order $2Nn+1$ (see [166, Sec. IV-A]). In other words, matrix (4.8) with $d=2Nn+1$ has full row rank, which implies

$$\tau \geqslant \tfrac{(2N\bar{n}+1)N\bar{m}}{T-2N\bar{n}} = \tfrac{16N^2+2N}{120-8N}.$$

In Fig. 4.3, we show the minimum number of input-output trajectories required to compute matrices in (4.33) in numerical simulations. The results tightly match the aforementioned two lower bounds, and the number of trajectories required by Corollary 4.2 is one order of magnitude less than that of [166, Thm. 1] when $N=14$.

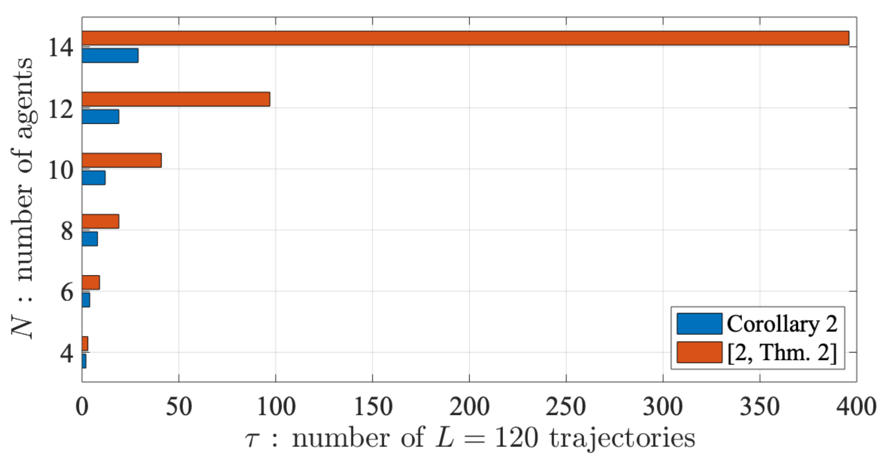


Figure 4.3: Minimum number of length $T=120$ input-output trajectories required to identify system (4.33).

4.3 *Related work and remarks*

The original Willems' lemma was proved in the context of behavior approach [182]. The lemma in [182] requires (A, B) in system (4.4) to be controllable, *i.e.*, $\mathcal{R} = \mathbb{R}^n$. In addition, the order of persistent excitation needed is $n+L$, and it is required for a single data trajectory. Later, van Waard *et al.* extend this result to multiple data trajectories. Such extension is particularly useful when there is missing data in a single data trajectory, or the data generating system is unstable, in which case obtaining multiple short trajectories will be more realistic than one single long trajectory. The generalization of Willems' lemma from controllable to possibly uncontrollable system was first introduced in [123]. The authors showed that any trajectories originated from the controllable subspace can be parameterized as a linear combination of data trajectories, which implies that the system's controllable sub-behavior is always identifiable. This point is further discussed in [115].

By parameterizing trajectories of the system (4.4) using data, the lemma has profound implications in system identification [117, 87, 114], and inspired lots of recent work in data-driven control. A partial list of its applications includes data-driven simulation [117, 13], output matching [116], control by interconnection [118], set-invariance control [20], linear quadratic regulation [42, 43, 154], and predictive control [79, 39, 4, 12, 3].

Future work The results in this chapter reveals two hidden factors in the original Willems' fundamental lemma: the role of controllability and minimal polynomial. However, there are still several open questions remaining. What if the input-output data is corrupted by noise? There has been some recent results that uses extended Kalman filters and averaging to handle Gaussian noise when using WIllems' fundamental lemma [4], and some heuristics when the data is generated by nonlinear systems instead of linear ones [39]. However, a solid theoretical understanding of these extensions of Willems's fundamental lemma is still missing.

One of the most attractive feature of Willems' fundamental lemma is that, compared with other results in data-driven control [42, 167], it only requires persistent excitation conditions on the input data, which is particularly useful in practice since, unlike state or output data, input data does not depends on the system's response. Hence one future direction can be developing input-data-only (sufficient) conditions for data-driven control objectives. Another direction is developing data-generating algorithms that not only satisfy persistent excitation conditions but also respect physical or operational constraints on the systems, so that the system is not damaged during the data-generating process.

Bibliography

[1] Yves Achdou and Mathieu Laurière. Mean field type control with congestion. *Appl. Math. Optim.*, 73(3):393–418, 2016.

[2] Behçet Açıkmeşe, Milan Mandić, and Jason L Speyer. Decentralized observers with consensus filters for distributed discrete-time linear systems. *Automatica*, 50(4):1037–1052, 2014.

[3] Ahmed Allibhoy and Jorge Cortés. Data-based receding horizon control of linear network systems. *arXiv preprint arXiv:2003.09813*, 2020.

[4] D. Alpago, F. Dörfler, and J. Lygeros. An extended Kalman filter for data-enabled predictive control. *IEEE Control Systems Letters*, 4(4):994–999, 2020.

[5] Eitan Altman. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999.

[6] Jay E Aronson. A survey of dynamic network flows. *Ann. Oper. Res.*, 20(1):1–66, 1989.

[7] Heinz H Bauschke. *Projection algorithms and monotone operators*. PhD thesis, Theses (Dept. of Mathematics and Statistics)/Simon Fraser University, 1996.

[8] Heinz H Bauschke and Patrick L Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, volume 408. Springer, 2017.

[9] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.

[10] Martin Beckmann, Charles B McGuire, and Christopher B Winsten. *Studies in the Economics of Transportation.* Yale University Press, 1956.

[11] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[12] J. Berberich, J. Koehler, M. A. Muller, and F. Allgower. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, pages 1–1, 2020.

[13] Julian Berberich and Frank Allgöwer. A trajectory-based framework for data-driven system analysis and control. In *Eur. Control Conf.*, pages 1365–1370. IEEE, 2020.

[14] James Bergin and Dan Bernhardt. Anonymous sequential games with aggregate uncertainty. *J. Math. Econ.*, 21(6):543–562, 1992.

[15] James Bergin and Dan Bernhardt. Anonymous sequential games: existence and characterization of equilibria. *Econ. Theory*, 5(3):461–489, 1995.

[16] Dimitri P Bertsekas. *Neuro-Dynamic Programming*. Athena scientific Belmont, 1996.

[17] Dimitri P Bertsekas. *Network Optimization: Continuous and Discrete Models*. Citeseer, 1998.

[18] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[19] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Math. Oper. Res.*, 16(3):580–595, 1991.

[20] Andrea Bisoffi, Claudio De Persis, and Pietro Tesi. Data-based guarantees of set invariance properties. *arXiv preprint arXiv:1911.12293[eecs.SY]*, 2019.

[21] Haley A Blanchard and Ambrose A Adegbege. An SOR-like method for fast model predictive control. *IFAC-PapersOnLine*, 50(1):14418–14423, 2017.

[22] Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. Gusto: Guaranteed sequential trajectory optimization via sequential convex programming. In *Int. Conf. Robot. Autom.*, pages 6741–6747. IEEE, 2019.

[23] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[24] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM Rev.*, 46(4):667–689, 2004.

[25] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[26] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

[27] Mathias Bürger, Daniel Zelazo, and Frank Allgöwer. Duality and network theory in passivity-based cooperative control. *Automatica*, 50(8):2051–2061, 2014.

[28] S Di Cairano, Matthew Brand, and Scott A Bortoff. Projection-free parallel quadratic programming for linear model predictive control. *Int. J. Control*, 86(8):1367–1385, 2013.

[29] Dan Calderone and Shankar Sastry. Infinite-horizon average-cost markov decision process routing games. In *Proc. Int. Conf. Intell. Transp. Syst.*, pages 1–6. IEEE, 2017.

[30] Dan Calderone and Shankar Sastry. Markov decision process routing games. In *Proc. Int. Conf. Cyber-Phys. Syst.*, pages 273–279. ACM, 2017.

[31] Daniel Joseph Calderone. *Models of Competition for Intelligent Transportation Infrastructure: Parking, Ridesharing, and External Factors in Routing Decisions*. PhD Thesis, 2017.

[32] Pierre Cardaliaguet, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions. The master equation and the convergence problem in mean field games. *arXiv preprint arXiv:1509.02505*, 2015.

[33] Yair Censor and Arnold Lent. An iterative row-action method for interval convex programming. *J. Optim. Theory Appl.*, 34(3):321–353, 1981.

[34] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011.

[35] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Program.*, 159(1-2):253–287, 2016.

[36] Tsung-Hui Chang, Mingyi Hong, and Xiangfeng Wang. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Trans. Signal Process.*, 63(2):482–497, 2015.

[37] Michael Cohen, Jelena Diakonikolas, and Lorenzo Orecchia. On acceleration with noise-corrupted gradients. In *International Conference on Machine Learning*, pages 1019–1028, 2018.

[38] José R Correa and Nicolás E Stier-Moses. Wardrop equilibria. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[39] Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the deepc. In *Eur. Control Conf.*, pages 307–312. IEEE, 2019.

[40] Thuy V Dang, Keck Voon Ling, and Jan M Maciejowski. Embedded ADMM-based QP solver for MPC with polytopic constraints. In *Proc. Eur. Control Conf.*, pages 3446–3451. IEEE, 2015.

[41] George Bernard Dantzig, Steven F Maier, ZF Lansdown, et al. Application of decomposition to transportation network analysis. Technical report, United States. Dept. of Transportation. Office of the Secretary, 1976.

[42] Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Trans. Autom. Control*, 65(3):909–924, 2019.

[43] Claudio De Persis and Pietro Tesi. Low-complexity learning of linear quadratic regulators from noisy data. *arXiv preprint arXiv:2005.01082[eecs.SY]*, 2020.

[44] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multi-block ADMM with $O(1/k)$ convergence. *J. Sci. Comput.*, 71(2):712–736, 2017.

[45] Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM Journal on Optimization*, 29(1):660–689, 2019.

[46] Thinh T Doan, Subhonmesh Bose, D Hoa Nguyen, and Carolyn L Beck. Convergence of the iterates in mirror descent methods. *IEEE Control Syst. Lett.*, 3(1):114–119, 2019.

[47] Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP solver for embedded systems. In *Proc. Eur. Control Conf.*, pages 3071–3076. IEEE, 2013.

[48] Alexander Domahidi, Aldo U Zgraggen, Melanie N Zeilinger, Manfred Morari, and Colin N Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *Proc. IEEE Conf. Decision Control*, pages 668–674. IEEE, 2012.

[49] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *Proc. Int. Conf. Mach. Learn.*, pages 272–279, 2008.

[50] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control*, 57(3):592–606, 2012.

[51] John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Proc. Ann. Conf. Learn. Theory*, pages 14–26, 2010.

[52] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

[53] Mahmoud El Chamie, Yue Yu, and Behçet Açıkmeşe. Convex synthesis of randomized policies for controlled Markov chains with density safety upper bound constraints. In *Proc. Amer. Control Conf.*, pages 6290–6295, 2016.

[54] Mahmoud El Chamie, Yue Yu, Behcet Acikmese, and Masahiro Ono. Controlled markov processes with safety state constraints. *IEEE Trans. Autom. Control*, 2018.

[55] Utku Eren, Anna Prach, Başaran Bahadır Koçer, Saša V Raković, Erdal Kayacan, and Behçet Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *J. Guid. Control Dyn.*, 40(7):1541–1566, 2017.

[56] David Evangelista, Rita Ferreira, Diogo A Gomes, Levon Nurbekyan, and Vardan Voskanyan. First-order, stationary mean-field games with congestion. *Nonlinear Anal.*, 173:37–74, 2018.

[57] Maria Fonoberova. Algorithms for finding optimal flows in dynamic networks. In *Handbook of Power Systems II*, pages 31–54. Springer, 2010.

[58] Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[59] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 3(1-2):95–110, 1956.

[60] Masao Fukushima. On the dual approach to the traffic assignment problem. *Transp. Res. B*, 18(3):235–245, 1984.

[61] Nathan H Gartner. Optimal traffic assignment with elastic demands: A review, Part I. Analysis framework. *Transp. Sci.*, 14(2):174–191, 1980.

[62] Nathan H Gartner. Optimal traffic assignment with elastic demands: A review, Part II. Algorithmic approaches. *Transp. Sci.*, 14(2):192–208, 1980.

[63] Nicolas Gast, Bruno Gaujal, and Jean-Yves Le Boudec. Mean field for markov decision processes: from discrete to continuous optimization. *IEEE Trans. Autom. Control*, 57(9):2266–2280, 2012.

[64] Reza Ghaemi, Jing Sun, and Ilya V Kolmanovsky. An integrated perturbation analysis and sequential quadratic programming approach for model predictive control. *Automatica*, 45(10):2412–2418, 2009.

[65] Behnam Gholami, Sejong Yoon, and Vladimir Pavlovic. Decentralized approximate bayesian inference for distributed sensor network. In *AAAI Conf. Artificial Intell.*, pages 1582–1588, 2016.

[66] Pontus Giselsson. Improved fast dual gradient methods for embedded model predictive control. *IFAC Proc. Vol.*, 47(3):2303–2309, 2014.

[67] Diogo A Gomes, Joana Mohr, and Rafael R Souza. Discrete mean field games. *arXiv preprint arXiv:0903.1620*, 2009.

[68] Diogo A Gomes, Joana Mohr, and Rafael Rigao Souza. Discrete time, finite state space mean field games. *J. Math. Pures Appl.*, 93(3):308–328, 2010.

[69] Olivier Guéant. From infinity to one: The reduction of some mean field games to a global control problem. *arXiv preprint arXiv:1110.3441*, 2011.

[70] Olivier Guéant. Existence and uniqueness result for mean field games with congestion effect on graphs. *Appl. Math. Optim.*, 72(2):291–303, 2015.

[71] Olivier Guéant, Jean-Michel Lasry, and Pierre-Louis Lions. Mean field games and applications. In *Paris-Princeton Lectures on Mathematical Finance 2010*, pages 205–266. Springer, 2011.

[72] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2018.

[73] HW Hamacher. Mathematical modeling of evacuation problems: A state of the art. *Pedestrian and Evacuation Dynamics*, pages 227–266, 2002.

[74] Bingsheng He, Min Tao, and Xiaoming Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM J. Optim.*, 22(2):313–340, 2012.

[75] Bingsheng He and Xiaoming Yuan. On the o(1/n) convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50(2):700–709, 2012.

[76] Niao He, Anatoli Juditsky, and Arkadi Nemirovski. Mirror prox algorithm for multi-term composite minimization and semi-separable problems. *Comput. Optim. Appl.*, 61(2):275–319, 2015.

[77] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Math. Program.*, 162(1-2):165–199, 2017.

[78] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[79] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control for grid-connected power converters. In *IEEE Conf. Decision Control*, pages 8130–8135. IEEE, 2019.

[80] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *Proc. IEEE Conf. Decision Control*, pages 3671–3676, 2013.

[81] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proc. Int. Conf. Mach. Learn.*, pages 427–435, 2013.

[82] Anoop Jain, Miel Sharf, and Daniel Zelazo. Regularization and feedback passivation in cooperative control of passivity-short systems: A network optimization perspective. *IEEE Control Syst. Lett.*, 2(4):731–736, 2018.

[83] Dusan Jakovetic, José MF Moura, and João Xavier. Distributed augmented lagrangian algorithms: convergence rate. In *Proc. IEEE Global Conf. Signal Inform. Process.*, pages 563–566, 2013.

[84] Juan L Jerez, Paul J Goulart, Stefan Richter, George A Constantinides, Eric C Kerrigan, and Manfred Morari. Embedded online optimization for model predictive control at megahertz rates. *IEEE Trans. Automat. Control*, 59(12):3238–3251, 2014.

[85] Boyan Jovanovic and Robert W Rosenthal. Anonymous sequential games. *J. Math. Econ.*, 17(1):77–87, 1988.

[86] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[87] Tohru Katayama. *Subspace methods for system identification*. Springer Science & Business Media, 2006.

[88] Markus Kögel and Rolf Findeisen. Fast predictive control of linear systems combining Nesterov's gradient method and the method of multipliers. In *Proc. IEEE Conf. Decision Control and Eur. Control Conf.*, pages 501–506. IEEE, 2011.

[89] Ekkehard Köhler, Rolf H Möhring, and Martin Skutella. Traffic networks and flows over time. In *Algorithmics of Large and Complex Networks*, pages 166–196. Springer, 2009.

[90] GM Korpelevich. Extragradient method for finding saddle points and other problems. *Matekon*, 13(4):35–49, 1977.

[91] Samir Kouro, Patricio Cortés, René Vargas, Ulrich Ammann, and José Rodríguez. Model predictive control—-a simple and powerful method to control power converters. *IEEE Trans. Ind. Electron.*, 56(6):1826–1838, 2008.

[92] Dimitris Kouzoupis, Gianluca Frison, Andrea Zanelli, and Moritz Diehl. Recent advances in quadratic programming algorithms for nonlinear model predictive control. *Vietnam J. Math.*, 46(4):863–882, 2018.

[93] Walid Krichene and Peter L Bartlett. Acceleration and averaging in stochastic descent dynamics. In *Advances in Neural Information Processing Systems*, pages 6796–6806, 2017.

[94] Walid Krichene, Alexandre Bayen, and Peter L Bartlett. Accelerated mirror descent in continuous and discrete time. In *Advances in neural information processing systems*, pages 2845–2853, 2015.

[95] Walid Krichene, Syrine Krichene, and Alexandre Bayen. Efficient Bregman projections onto the simplex. In *Proc. IEEE Conf. Decision Control*, pages 3291–3298. IEEE, 2015.

[96] D Kwame Minde Kufoalor, Stefan Richter, Lars Imsland, Tor Arne Johansen, Manfred Morari, and Gisle Otto Eikrem. Embedded model predictive control on a PLC using a primal-dual first-order method for a subsea separation process. In *Proc. Mediterranean Conf. Control Autom.*, pages 368–373. IEEE, 2014.

[97] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an O(1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002[cs.LG]*, 2012.

[98] Torbjörn Larsson, Zhuangwei Liu, and Michael Patriksson. A dual scheme for traffic assignment problems. *Optimization*, 42(4):323–358, 1997.

[99] Torbjörn Larsson and Michael Patriksson. Side constrained traffic equilibrium models—analysis, computation and applications. *Transp. Res. B*, 33(4):233–264, 1999.

[100] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japan. J. Math.*, 2(1):229–260, 2007.

[101] Victor Lesser, Charles L Ortiz Jr, and Milind Tambe. *Distributed Sensor Networks: A Multiagent Perspective*, volume 9. Springer Science & Business Media, 2012.

[102] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*, 17(1):1334–1373, 2016.

[103] Dan Li, Kerry D Wong, Yu Hen Hu, and Akbar M Sayeed. Detection, classification, and tracking of targets. *IEEE Signal Process. Mag.*, 19(2):17–29, 2002.

[104] Jueyou Li, Guo Chen, Zhaoyang Dong, and Zhiyou Wu. Distributed mirror descent method for multi-agent optimization with delay. *Neurocomputing*, 177:643–650, 2016.

[105] Jueyou Li, Guoquan Li, Zhiyou Wu, and Changzhi Wu. Stochastic mirror descent method for distributed multi-agent optimization. *Optim. Lett.*, 12(6):1179–1197, 2018.

[106] Sarah HQ Li, Yue Yu, Daniel Calderone, Lillian Ratliff, and Behçet Açrkmeşe. Tolling for constraint satisfaction in markov decision process congestion games. In *2019 American Control Conference (ACC)*, pages 1238–1243. IEEE, 2019.

[107] Dominic Liao-McPherson, Mike Huang, and Ilya Kolmanovsky. A regularized and smoothed Fischer–Burmeister method for quadratic programming with applications to model predictive control. *IEEE Trans. Automat. Control*, 64(7):2937–2944, 2018.

[108] Qing Ling and Alejandro Ribeiro. Decentralized dynamic optimization through the alternating direction method of multipliers. In *Proc. IEEE Workshop Signal Process. Adv. Wireless Communn.*, pages 170–174, 2013.

[109] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD Conf.*, 2004.

[110] Ali Makhdoumi and Asuman Ozdaglar. Convergence rate of distributed ADMM over networks. *IEEE Trans. Autom. Control*, 62(10):5082–5095, 2017.

[111] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems with state constraints. *IFAC-PapersOnLine*, 50(1):4063–4069, 2017.

[112] Yuanqi Mao, Michael Szmuk, and Behçet Açıkmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. In *IEEE Conf. Decision Control*, pages 3636–3641. IEEE, 2016.

[113] Yuanqi Mao, Michael Szmuk, Xiangru Xu, and Behçet Açikmese. Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems. *arXiv preprint arXiv:1804.06539[math.OC]*, 2018.

[114] Ivan Markovsky. From data to models. In *Low-Rank Approximation*, pages 37–70. Springer, 2019.

[115] Ivan Markovsky and Florian Dörfler. Identifiability in the behavioral setting. Technical report, Dept. ELEC, Vrije Universiteit Brussel, 2020.

[116] Ivan Markovsky and Paolo Rapisarda. Data-driven simulation and control. *Int. J. Control*, 81(12):1946–1959, 2008.

[117] Ivan Markovsky, Jan C Willems, Paolo Rapisarda, and Bart LM De Moor. Algorithms for deterministic balanced subspace identification. *Automatica*, 41(5):755–766, 2005.

[118] TM Maupong and Paolo Rapisarda. Data-driven control: A behavioral approach. *Syst. Control Lett.*, 101:37–43, 2017.

[119] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[120] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[121] De Meng, Maryam Fazel, and Mehran Mesbahi. Proximal alternating direction method of multipliers for distributed optimization on weighted graphs. In *Proc. IEEE Conf. Decision Control*, pages 1396–1401. IEEE, 2015.

[122] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[123] Vikas Kumar Mishra, Ivan Markovsky, and Ben Grossmann. Data-driven tests for controllability. *IEEE Control Syst. Lett.*, 5(2):517–522, 2020.

[124] John D Murchland. Road network traffic distribution in equilibrium. In *Proc. Conf. Math. Methods Econ. Sci.*, 1969.

[125] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61, 2009.

[126] Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *J. Optim. Theory Appl.*, 142(1):205–228, 2009.

[127] Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control*, 55(4):922–938, 2010.

[128] Arkadi Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim*, 15(1):229–251, 2004.

[129] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2009.

[130] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley and Sons, 1983.

[131] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2010.

[132] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

[133] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5):780–798, 2005.

[134] Brendan O'Donoghue, Giorgos Stathopoulos, and Stephen Boyd. A splitting method for optimal control. *IEEE Trans. Control Syst. Technol.*, 21(6):2432–2442, 2013.

[135] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 49(9):1520–1533, 2004.

[136] Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Math. Program.*, pages 1–35, 2019.

[137] Michael Patriksson. *The Traffic Assignment Problem: Models and Methods.* Courier Dover Publications, 1994.

[138] Panagiotis Patrinos and Alberto Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans. Automat. Control*, 59(1):18–33, 2013.

[139] Panagiotis Patrinos, Pantelis Sopasakis, and Haralambos Sarimveis. A global piecewise smooth Newton method for fast large-scale model predictive control. *Automatica*, 47(9):2016–2022, 2011.

[140] Warren B Powell, Patrick Jaillet, and Amedeo Odoni. Stochastic and dynamic networks and routing. *Handbooks in Operations Research and Management Science*, 8:141–295, 1995.

[141] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[142] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control. Eng. Pract.*, 11(7):733–764, 2003.

[143] Zhirong Qiu, Shuai Liu, and Lihua Xie. Distributed constrained optimal consensus of multi-agent systems. *Automatica*, 68:209–215, 2016.

[144] Maxim Raginsky and Jake Bouvrie. Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence. In *Proc. Conf. Decision Control*, pages 6793–6800. IEEE, 2012.

[145] James Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Math. Prog.*, 40(1-3):59–93, 1988.

[146] Felix Rey, Damian Frick, Alexander Domahidi, Juan Jerez, Manfred Morari, and John Lygeros. ADMM prescaling for model predictive control. In *Proc. IEEE Conf. Decision Control*, pages 3662–3667. IEEE, 2016.

[147] Stefan Richter, Colin Neil Jones, and Manfred Morari. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Trans. Automat. Control*, 57(6):1391–1403, 2011.

[148] Stefan Richter, Colin Neil Jones, and Manfred Morari. Certification aspects of the fast gradient method for solving the dual of parametric convex programs. *Math. Meth. Oper. Res.*, 77(3):305–321, 2013.

[149] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[150] R Tyrrell Rockafellar. *Network Flows and Monotropic Optimization*. John Wiley & Sons, 1984.

[151] R Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

[152] Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[153] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton University Press, 2015.

[154] Monica Rotulo, Claudio De Persis, and Pietro Tesi. Data-driven linear quadratic regulation via semidefinite programming. *arXiv preprint arXiv:1911.07767*, 2019.

[155] Tim Roughgarden. Routing games. *Algorithmic Game Theory*, 18:459–484, 2007.

[156] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l_1-regularized loss minimization. *J. Mach. Learn. Res.*, 12(Jun):1865–1892, 2011.

[157] Miel Sharf and Daniel Zelazo. A network optimization approach to cooperative control synthesis. *IEEE Control Syst. Lett.*, 1(1):86–91, 2017.

[158] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans. Signal Process.*, 62(7):1750–1761, 2014.

[159] Martin Skutella. An introduction to network flows over time. In *Research Trends in Combinatorial Optimization*, pages 451–482. Springer, 2009.

[160] Leo Emil Sokoler, Gianluca Frison, Martin S Andersen, and John Bagterp Jørgensen. Input-constrained model predictive control via the alternating direction method of multipliers. In *Proc. Eur. Control Conf.*, pages 115–120. IEEE, 2014.

[161] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pages 2510–2518, 2014.

[162] Akhil Sundararajan, Bryan Van Scoy, and Laurent Lessard. Analysis and design of first-order distributed optimization algorithms over time-varying graphs. *IEEE Transactions on Control of Network Systems*, 2020.

[163] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 2154–2162, 2016.

[164] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Math. Prog.*, 125(2):263–295, 2010.

[165] Fabian Ullmann. Fiordos: A matlab toolbox for c-code generation for first order methods. *Master thesis*, 2011.

[166] Henk J van Waarde, Claudio De Persis, M Kanat Camlibel, and Pietro Tesi. Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Syst. Lett.*, 4(3):602–607, 2020.

[167] Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: a new perspective on data-driven analysis and control. *IEEE Trans. Autom. Control*, 2020.

[168] Michel Verhaegen and Vincent Verdult. *Filtering and system identification: a least squares approach*. Cambridge University Press, 2007.

[169] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1953.

[170] Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of multi-block bregman ADMM for nonconvex composite problems. *arXiv preprint arXiv:1505.03063*, 2015.

[171] Huahua Wang and Arindam Banerjee. Bregman alternating direction method of multipliers. In *Adv. Neural Inform. Process. Syst.*, pages 2816–2824, 2014.

[172] Huahua Wang, Arindam Banerjee, and Zhi-Quan Luo. Parallel direction method of multipliers. In *Adv. Neural Inform. Process. Syst.*, pages 181–189, 2014.

[173] Jing Wang and Nicola Elia. Control approach to distributed optimization. In *Proc. Allerton Conf. Commun. Control Comput.*, pages 557–561. IEEE, 2010.

[174] Jing Wang and Nicola Elia. A control perspective for centralized and distributed convex optimization. In *Proc. IEEE Conf. Decision Control*, pages 3800–3805. IEEE, 2011.

[175] Yang Wang and Stephen Boyd. Fast model predictive control using online optimization. *IEEE Trans. Control Syst. Technol.*, 18(2):267–278, 2009.

[176] Yinghui Wang, Hongbing Zhou, and Yiguang Hong. Distributed stochastic mirror descent algorithm over time-varying network. In *Proc. Int. Conf. Control Automat.*, pages 716–721. IEEE, 2018.

[177] John Glen Wardrop and James Ivor Whitehead. Correspondence. some theoretical aspects of road traffic research. *Proc. Inst. Civil Eng.*, 1(5):767–768, 1952.

[178] Ermin Wei and Asuman Ozdaglar. Distributed alternating direction method of multipliers. In *Proc. IEEE Conf. Decision Control*, pages 5445–5450. IEEE, 2012.

[179] Guanghui Wen, Zhisheng Duan, Wei Ren, and Guanrong Chen. Distributed consensus of multi-agent systems with general linear node dynamics and intermittent communications. *Int. J. Robust. Nonlinear Control*, 24(16):2438–2457, 2014.

[180] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proc. Nat. Acad. of Sci.*, 113(47):E7351–E7358, 2016.

[181] Piotr Więcek and Eitan Altman. Stationary anonymous sequential games with undiscounted rewards. *J. Optim. Theory Appl.*, 166(2):686–710, 2015.

[182] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Syst. Control Lett.*, 54(4):325–329, 2005.

[183] C Xi, Q Wu, and UA Khan. Distributed mirror descent over directed graphs. *arXiv preprint arXiv:1412.5526[math.OC]*, 2014.

[184] Lin Xiao, Stephen Boyd, and Seung-Jean Kim. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67(1):33–46, 2007.

[185] Lin Xiao, Mikael Johansson, and Stephen P Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun.*, 52(7):1136–1144, 2004.

[186] Yangyang Xu. Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM J. Optim.*, 27(3):1459–1484, 2017.

[187] Ali Yahya, Adrian Li, Mrinal Kalakrishnan, Yevgen Chebotar, and Sergey Levine. Collective robot reinforcement learning with distributed asynchronous guided policy search. In *Int. Conf. Intell. Robots Syst.*, pages 79–86. IEEE, 2017.

[188] Shaofu Yang, Qingshan Liu, and Jun Wang. A multi-agent system with a proportional-integral protocol for distributed constrained optimization. *IEEE Trans. Autom. Control*, 62(7):3461–3467, 2017.

[189] Yue Yu and Behçet Açıkmeşe. Stochastic Bregman parallel direction method of multipliers for distributed optimization. *arXiv preprint arXiv:1902.09695[math.OC]*, 2019.

[190] Yue Yu and Behçet Açıkmeşe. RLC circuits-based distributed mirror descent method. *IEEE Control Syst. Lett.*, 4(3):548–553, 2020.

[191] Yue Yu, Behçet Açıkmeşe, and Mehran Mesbahi. Bregman parallel direction method of multipliers for distributed optimization via mirror averaging. *IEEE Control Syst. Lett.*, 2(2):302–306, 2018.

[192] Yue Yu, Behçet Açıkmeşe, and Mehran Mesbahi. Mass–spring–damper networks for distributed optimization in non-Euclidean spaces. *Automatica*, 112:108703, 2020.

[193] Yue Yu, Dan Calderone, Sarah HQ Li, Lillian J Ratliff, and Behçet Açıkmeşe. A primal-dual approach to markovian network optimization. *arXiv preprint arXiv:1901.08731[math.OC]*, 2019.

[194] Yue Yu, Purnanand Elango, and Behçet Açikmeşe. Proportional-integral projected gradient method for model predictive control. *arXiv preprint arXiv:2009.06980*, 2020.

[195] Deming Yuan, Yiguang Hong, Daniel WC Ho, and Guoping Jiang. Optimal distributed stochastic mirror descent for strongly convex optimization. *Automatica*, 90:196–203, 2018.

[196] Costantinos Zagaris, Hyeongjun Park, Josep Virgili-Llop, Richard Zappulla, Marcello Romano, and Ilya Kolmanovsky. Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints. *Journal of Guidance, Control, and Dynamics*, 41(9):2054–2062, 2018.

[197] Ruohan Zhang, Yue Yu, Mahmoud El Chamie, Behçet Açikmese, and Dana H Ballard. Decision-making policies for heterogeneous autonomous multi-agent systems with safety constraints. In *Proc. Int. Joint Conf. Artificial Intell.*, pages 546–553, 2016.

[198] Pablo Zometa, Markus Kögel, and Rolf Findeisen. μAO-MPC: a free code generation tool for embedded real-time linear model predictive control. In *2013 American Control Conference*, pages 5320–5325. IEEE, 2013.

A Convex analysis and optimization

This section introduces some basic results in convex analysis and optimization. We refer interested readers to [152] for a detailed discussion.

A.1 Convex functions

In this section, we introduce some basic results about convex sets and functions.

We say set $\mathbb{X} \subset \mathbb{R}^n$ is closed if it contains all its limit points, we say $\mathbb{X}$ is convex if

$$\alpha x + (1-\alpha)x' \in \mathbb{X}, \;\; \forall x, x' \in \mathbb{X}, \alpha \in [0,1] \tag{A.1}$$

For a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, we denote $f : \mathbb{X} \to \mathbb{R}$ if

$$\mathbb{X} = \{x | f(x) < \infty\}. \tag{A.2}$$

We also define its *epigraph* as

$$\mathbf{epi}\, f = \{(x, \alpha) | x \in \mathbb{X}, \alpha \geqslant f(x)\}. \tag{A.3}$$

Epigraphs provide a mapping between a function and a set. As a result, we can define many useful properties of functions using corresponding properties on sets. In particular, we say

- function $f : \mathbb{X} \to \mathbb{R}$ is convex if $\mathbf{epi}\, f$ is a convex set
- function $f : \mathbb{X} \to \mathbb{R}$ is proper if $\mathbf{epi}\, f$ is non-empty and contains no vertical lines
- if function $f : \mathbb{X} \to \mathbb{R}$ is convex and proper, then function f is closed if $\mathbf{epi}\, f$ is a closed set.

If function $f : \mathbb{X} \to \mathbb{R}$ is convex, the following Jensen's inequality holds

$$f\left(\sum_{i=1}^{N} \frac{\alpha^i}{\sum_{i=1}^{N} \alpha^i} x^i\right) \leqslant \frac{1}{\sum_{i=1}^{N} \alpha^i} \sum_{i=1}^{N} \alpha^i f(x^i), \quad \forall \alpha^1, \ldots, \alpha^N \in \mathbb{R}_+ \tag{A.4}$$

For a closed convex proper function $f : \mathbb{X} \to \mathbb{R}$, its subdifferential at $x \in \mathbb{X}$ is defined as

$$\partial f(x) = \{u | f(x') - f(x) \geqslant \langle u, x' - x\rangle, \forall x' \in \mathbb{X}\}. \tag{A.5}$$

We say $u \in \mathbb{R}^n$ is a subgradient of function $f : \mathbb{X} \to \mathbb{R}$ at $x \in \mathbb{X}$ if

$$u \in \partial f(x). \tag{A.6}$$

If, in addition, that function f is continuously differentiable over $\mathbb{X}$, then its subdifferential reduces to a singleton, *i.e.*,

$$\partial f(x) = \{\nabla f(x)\}. \tag{A.7}$$

The following theorem shows one of the fundamental property of subgradients.

Theorem A.1 (Thm. 23.5, [152])**.** *If $f : \mathbb{X} \to \mathbb{R}$ is a closed convex proper function, then, for any $x \in \mathbb{X}$ and $u \in \partial f(x)$,*

$$x \in \operatorname*{argmax}_{y \in \mathbb{X}} \ \langle y, u\rangle - f(y).$$

A.2 Optimality conditions

In this section, we discuss the conditions that characterize the minimizers of a convex objective function over a convex constraint set. Such optimality conditions change with the descriptions of the constraint set. In the following, we consider three representative cases.

A.2.1 Abstract set constraint

Consider the following optimization problem with abstract set constraints

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathbb{X} \end{aligned} \tag{A.8}$$

where set $\mathbb{X} \subset \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ are both convex. The following theorem gives the optimality conditions of problem (A.8).

Theorem A.2 (Thm. 27.4, [152])**.** *Consider optimization problem* (A.8)*. Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is proper and convex, set $\mathbb{X}$ closed and convex. Then x is an optimal solution to optimization* (A.8) *if and only if there exists $u \in \partial f(x)$ such that $\langle u, x' - x \rangle \geqslant 0$ for all $x' \in \mathbb{X}$.*

The following lemma shows the an important special case of Theorem A.2 where the optimizer is the Euclidean projection projection of a reference point

Lemma A.3. *[131, Lemma. 2.2.7] Let set $\mathbb{X} \subset \mathbb{R}^n$ be a closed and convex, $x \in \mathbb{R}^n$ be an arbitrary vector, and*

$$\pi_{\mathbb{X}}[x] = \underset{y \in \mathbb{X}}{\operatorname{argmin}} \, \|y - x\|_2 = \underset{y \in \mathbb{X}}{\operatorname{argmin}} \, \frac{1}{2} \|y - x\|_2^2 \, , .$$

Then

$$\langle \pi_{\mathbb{X}}[x] - x, x' - \pi_{\mathbb{X}}[x] \rangle \geqslant 0, \;\; \forall x \in \mathbb{R}^n, x' \in \mathbb{X}.$$

A.2.2 Affine constraints

Consider the following optimization problem with linear constraints

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & Ax = b, \;\; x \geqslant 0 \end{aligned} \tag{A.9}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. We define the *Lagrangian* of (A.9) as

$$L(x, u, v) = f(x) + \langle Ax - b, v\rangle - \langle x, u\rangle \tag{A.10}$$

where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are also known as *Lagrangian multipliers.* The Karush–Kuhn–Tucker (KKT) conditions of Lagrangian (A.10) are given by

$$\begin{aligned} Ax = b,\ \ x \geqslant& 0, & \text{(A.11a)}\\ u \geqslant& 0, & \text{(A.11b)}\\ w + A^\top v - u =& 0,\ \ w \in \partial f(x) & \text{(A.11c)}\\ x_i u_i =& 0,\ \ i = 1, 2, \ldots, n. & \text{(A.11d)} \end{aligned}$$

where x_i and u_i are the i-th element of vector $x \in \mathbb{R}^n$ and, respectively, $u \in \mathbb{R}^n$. We will use the following KKT theorem for optimization (A.9).

Theorem A.4 (Cor. 28.3.1, [152])**.** *Consider optimization problem* (A.9)*. Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ convex, finite valued, and convex. Then x is an optimal solution to optimization* (A.9) *if and only if there exists $u \in \mathbb{R}^n, v \in \mathbb{R}^m$ and $w \in \partial f(x)$ such that conditions in* (A.11) *hold.*

A.2.3 Abstract set and affine constraints

Consider the following optimization with both abstract set constraints and affine constraints

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b,\ \ x \in \mathbb{X} \end{array} \tag{A.12}$$

Notice that the constraints in (A.12) combine those in (A.8) and (A.9). The following theorem gives the optimality conditions of problem (A.12).

Theorem A.5 (Thm. 3.1.27, [131])**.** *Consider optimization problem* (A.12)*. Suppose set*

$\mathbb{X}$ is closed and convex, function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, proper, and its level sets on $\mathbb{X}$ are bounded. Suppose there exists $\bar{x} \in \mathbb{R}^n$ and $\epsilon > 0$ such that

$$A\bar{x} = b, \quad \{x \in \mathbb{R}^n | \, \|x - \bar{x}\|_2 \leqslant \epsilon\} \subseteq \mathbb{X}$$

Then x is an optimal solution to (A.12) if and only if there exists $u \in \mathbb{R}^n$ and $w \in \partial f(x)$ such that

$$\begin{aligned} Ax =& b, \quad x \in \mathbb{X} \\ \langle w + A^\top u, x' - x \rangle \geqslant& 0, \quad \forall x' \in \mathbb{X}. \end{aligned}$$

B First order optimization methods

In this section, we will take a brief tour in the theory of first order methods of convex optimization. We will review some of the most influential algorithms in this domain, each paired with self-contained proofs of its convergence properties under different assumptions. We will use some of the results on convex analysis in Appendix A.

We consider the following constrained optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathbb{X} \end{aligned} \tag{B.1}$$

where

- function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper
- set $\mathbb{X} \subseteq \mathbb{R}^n$ is closed and convex

We will be interested in first order optimization methods for optimization (B.1), *i.e.*, methods that only uses first order information of function f, such as gradients or subgradients.

Using Theorem A.2, we can obtain a characterization of the optimizers to problem (B.1). In particular,

- If f is continuously differentiable, then $x^\star$ is an optimizer to problem (B.1) if and only if
$$\langle \nabla f(x^\star), x - x^\star \rangle \geqslant 0, \;\; \forall x \in \mathbb{X}. \tag{B.2}$$
- If f is not differentiable, then then $x^\star$ is an optimizer to problem (B.1) if and only if there exists $u^\star \in \partial f(x^\star)$
$$\langle u^\star, x - x^\star \rangle \geqslant 0, \;\; \forall x \in \mathbb{X}. \tag{B.3}$$

Our later analysis will heavily use of the following notion of Bregman divergence.

- If f is continuously differentiable, then we define the Bregman divergence from x' to x associated with f as

$$B_f(x', x) = f(x') - f(x) - \langle \nabla f(x), x' - x \rangle. \tag{B.4}$$

Using (B.4) one can verify

$$B_f(x', x) - B_f(x', x^+) - B_f(x^+, x) = \langle \nabla f(x^+) - \nabla f(x), x' - x^+ \rangle, \ \ \forall x, x', x^+. \tag{B.5}$$

If $f = \frac{1}{2}\left\|\cdot\right\|_2^2$, the above identity reduces to the famous *law of consines* as follows

$$\frac{1}{2}\left\|x' - x\right\|_2^2 - \frac{1}{2}\left\|x' - x^+\right\|_2^2 - \frac{1}{2}\left\|x^+ - x\right\|_2^2 = \langle x^+ - x, x' - x^+ \rangle, \ \ \forall x, x', x^+. \tag{B.6}$$

- If f is not differentiable, then we define the Bregman divergence from x' to x associated with f and subgradient $u \in \partial f(x)$ as

$$B_f(x', x; u) = f(x') - f(x) - \langle u, x' - x \rangle. \tag{B.7}$$

Using (B.7) one can verify

$$\begin{aligned} & B_f(x', x; u) - B_f(x', x^+; u^+) - B_f(x^+, x; u) \\ =& \langle u^+ - u, x' - x^+ \rangle, \ \ \forall x, x', x^+, \ \ u \in \partial f(x), u^+ \in \partial f(x^+) \end{aligned} \tag{B.8}$$

We will also use the following notion of Eulidean projection

$$\pi_{\mathbb{X}}[x] = \operatorname*{argmin}_{y \in \mathbb{X}} \left\|y - x\right\|_2 = \operatorname*{argmin}_{y \in \mathbb{X}} \frac{1}{2}\left\|y - x\right\|_2^2, . \tag{B.9}$$

In the following sections, we will discuss the convergence properties of several popular first order algorithms. Common to most of our convergence proofs[1] is the following **4-step formula**

- Step 1, optimality conditions of the current iterate and the the optimizers. This step usually uses Theorem A.2. After this step, one end up with a non-negative inner product.

- Step 2, identities of Bregman divergence. This step mostly uses (B.5) or (B.8). After this step, the inner product obtain from the previous step can (almost always) be replaced with Bregman divergence terms.

- Step 3, upper & lower bound on the Bregman divegence. This step cleans up the results from the previous step by bounding and canceling some Bregman divergence terms, and usually end up with a telescope series.

- Step 4, Jensen's inequality. This step uses (A.4), and transform the telescope series in the previous step into a convergence rate.

In the following, we will reuse the results discussed above (all results appeared in the above recipe), so we strongly encourage the readers to reread them before dive into a new proof. Other than that, each individual section is independent from each other, hence the readers can easily skip any proof if they prefer. Finally, compared with results in other sections, the accelerated mirror descent method discussed in Section B.6 is more recent [37], and its proof uses a recently developed technique called approximate duality gap theory [45]. The author is particularly fond of this technique and strongly encourage the readers to take a look at §B.6 and the corresponding references.

[1]The proofs in §B.5 and §B.6 do not use this formula.

B.1 Projected gradient method

If f is continuously differentiable and convex, then perhaps the simplest and most popular algorithm for optimization (B.1) is the projected gradient descent method, which iterates as follows

$$x^{k+1} = \pi_{\mathbb{X}}[x^k - \alpha \nabla f(x^k)] \tag{B.10}$$

In the remainder of this section, we will discuss the convergence property of the algorithm in (B.10) under different assumptions.

B.1.1 Non-strongly convex case

We start with the case where $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth and convex, but not necessarily strongly convex. Using the notion of Bregman divergence, this is to say that

$$0 \leqslant B_f(x', x) \leqslant \frac{\beta}{2} \|x' - x\|_2^2, \ \ \forall x, x' \in \mathbb{R}^n \tag{B.11}$$

With these additional assumption, the convergence property of (B.10) is as follows.

Theorem B.1. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, continuously differentiable, and* (B.11) *holds. Let sequence $\{x^j\}$ be generated by* (B.10) *with $\alpha \leqslant \frac{1}{\beta}$, $x^\star$ be an optimizer of optimization* (B.1), *and $\overline{x}^k = \frac{1}{k}\sum_{j=1}^k x^{j+1}$. Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{\|x^1 - x^\star\|_2^2}{2\alpha k}$$

Proof. Step 1, applying Lemma A.3 to (B.10) gives the following

$$0 \leqslant \langle x^{j+1} - (x^j - \alpha \nabla f(x^j)), x^\star - x^{j+1} \rangle \tag{B.12}$$

Adding $\alpha\langle \nabla f(x^\star), x^{j+1} - x^\star \rangle$ to the both sides of the above inequality gives

$$\alpha\langle \nabla f(x^\star), x^{j+1} - x^\star \rangle \leqslant \langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle + \alpha\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle \quad \text{(B.13)}$$

Step 2, one can verify the following using (B.5)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle = \frac{1}{2}\left\| x^j - x^\star \right\|_2^2 - \frac{1}{2}\left\| x^{j+1} - x^\star \right\|_2^2 - \frac{1}{2}\left\| x^{j+1} - x^j \right\|_2^2 \quad \text{(B.14a)}$$

$$\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle = B_f(x^{j+1}, x^j) - B_f(x^{j+1}, x^\star) - B_f(x^\star, x^j) \quad \text{(B.14b)}$$

Step 3, since f is β-smooth and convex, we know that

$$B_f(x^{j+1}, x^j) \leqslant \frac{\beta}{2}\left\| x^{j+1} - x^j \right\|_2^2, \quad \text{(B.15a)}$$

$$0 \leqslant B_f(x^\star, x^j). \quad \text{(B.15b)}$$

Summing up (B.13), (B.14a), $\alpha\times$(B.14b), $\alpha\times$(B.15a) and $\alpha\times$(B.15b), and use the assumption that $\alpha \leqslant \frac{1}{\beta}$, we obtain the following

$$\alpha(f(x^{j+1}) - f(x^\star)) \leqslant \frac{1}{2}\left\| x^j - x^\star \right\|_2^2 - \frac{1}{2}\left\| x^{j+1} - x^\star \right\|_2^2 \quad \text{(B.16)}$$

where we also used the fact that $B_f(x^{j+1}, x^\star) = f(x^{j+1}) - f(x^\star) - \langle \nabla f(x^\star), x^{j+1} - x^\star \rangle$. Summing up (B.16) from $j = 1$ to $j = k$ we obtain

$$\alpha \sum_{j=1}^{k} (f(x^{j+1}) - f(x^\star)) \leqslant \frac{1}{2}\left\| x^1 - x^\star \right\|_2^2 - \frac{1}{2}\left\| x^{j+1} - x^\star \right\|_2^2 \leqslant \frac{1}{2}\left\| x^1 - x^\star \right\|_2^2. \quad \text{(B.17)}$$

Step 4, since f is convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. □

B.1.2 Strongly convex case

What happens if we strengthen the assumptions in (B.11) by assuming $f : \mathbb{R}^n \to \mathbb{R}$ to be both smooth and strongly convex? *i.e.*,

$$\frac{\mu}{2} \|x' - x\|_2^2 \leqslant B_f(x', x) \leqslant \frac{\beta}{2} \|x' - x\|_2^2, \ \ \forall x, x' \in \mathbb{R}^n \tag{B.18}$$

The following theorem shows that, with the above assumptions, algorithm (B.10) ensures the quadratic distance to optimum shrinks exponentially.

Theorem B.2. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, continuously differentiable, and* (B.18) *holds. Let sequence $\{x^j\}$ be generated by* (B.10) *with $\alpha = \frac{1}{\beta}$, and $x^\star$ be an optimizer of optimization* (B.1)*. Then*

$$\left\|x^{k+1} - x^\star\right\|_2^2 \leqslant \left(\frac{\beta - \mu}{\beta + \mu}\right)^k \left\|x^1 - x^\star\right\|_2^2$$

Proof. Step 1, applying Lemma A.3 to (B.10), and letting $x = x^{j+1}$ in (B.2) gives the following

$$0 \leqslant \langle x^{j+1} - (x^j - \alpha \nabla f(x^j)), x^\star - x^{j+1} \rangle \tag{B.19a}$$
$$0 \leqslant \langle \nabla f(x^\star), x^{j+1} - x^\star \rangle \tag{B.19b}$$

Multiplying the second inequality with α then adding it to the first one gives

$$0 \leqslant \langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle + \alpha \langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle \tag{B.20}$$

Step 2, one can verify the following using (B.5)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle = \frac{1}{2} \left\|x^j - x^\star\right\|_2^2 - \frac{1}{2} \left\|x^{j+1} - x^\star\right\|_2^2 - \frac{1}{2} \left\|x^{j+1} - x^j\right\|_2^2 \tag{B.21a}$$
$$\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle = B_f(x^{j+1}, x^j) - B_f(x^{j+1}, x^\star) - B_f(x^\star, x^j) \tag{B.21b}$$

Step 3, since f is β-smooth and μ-strongly convex, we know that

$$B_f(x^{j+1}, x^j) \leqslant \frac{\beta}{2} \left\|x^{j+1} - x^j\right\|_2^2, \tag{B.22a}$$

$$\frac{\mu}{2} \left\|x^j - x^\star\right\|_2^2 \leqslant B_f(x^\star, x^j), \tag{B.22b}$$

$$\frac{\mu}{2} \left\|x^{j+1} - x^\star\right\|_2^2 \leqslant B_f(x^{j+1}, x^\star). \tag{B.22c}$$

Summing up (B.20), (B.21a), $\alpha\times$(B.21b), $\alpha\times$(B.22a), $\alpha\times$(B.22b) and $\alpha\times$(B.22c), we obtain the following

$$\frac{1+\mu/\beta}{2} \left\|x^{j+1} - x^\star\right\|_2^2 \leqslant \frac{1-\mu/\beta}{2} \left\|x^j - x^\star\right\|_2^2 \tag{B.23}$$

where we use the assumption that $\alpha = \frac{1}{\beta}$. Using the above inequality recursively for k times, we obtain the desired results. □

B.2 Projected subgradient method

If $f : \mathbb{R}^n \to \mathbb{R}$ is convex but not necessarily differentiable, then problem (B.1) can be solved using a variant of (B.10), named projected subgradient method, which iterates as follows

$$x^{k+1} = \pi_{\mathbb{X}}[x^k - \alpha^k u^k], \;\; u^k \in \partial f(x^k) \tag{B.24}$$

In the remainder of this section, we will discuss the convergence properties of (B.24) under different assumptions.

B.2.1 Non-strongly convex case

We start with the case where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and γ-Lipschitz. Using the notion of Bregman divergence, this is to say that

$$0 \leqslant B_f(x', x; u), \;\; \forall x, x' \in \mathbb{R}^n, \;\; u \in \partial f(x) \tag{B.25a}$$

$$\|u\|_2 \leqslant \gamma, \;\; \forall u \in \partial f(x), x \in \mathbb{R}^n \tag{B.25b}$$

With these additional assumption, the convergence property of (B.24) is as follows.

Theorem B.3. *Suppose function* $f : \mathbb{R}^n \to \mathbb{R}$ *is closed, convex, proper, and* (B.25) *holds. Let sequence* $\{x^j\}$ *be generated by* (B.24), $x^\star$ *be an optimizer of optimization* (B.1), *and* $\overline{x}^k = \frac{1}{\sum_{j=1}^k \alpha^j} \sum_{j=1}^j \alpha^j x^j$. *Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{\|x^1 - x^\star\|_2^2 + \gamma^2 \sum_{j=1}^k (\alpha^j)^2}{2\sum_{j=1}^k \alpha^j}$$

Proof. Step 1, applying Lemma A.3 to (B.24) gives

$$0 \leqslant \langle x^{j+1} - (x^j - \alpha^j u^j), x^\star - x^{j+1} \rangle \tag{B.26}$$

Adding $\alpha^j \langle u^\star, x^j - x^\star \rangle$ to the both sides of the above inequality gives

$$\alpha^j \langle u^\star, x^j - x^\star \rangle \leqslant \langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle + \alpha^j \langle u^\star - u^j, x^j - x^\star \rangle + \alpha^j \langle u^j, x^j - x^{j+1} \rangle \tag{B.27}$$

Step 2, one can verify the following using (B.5) and (B.8)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1} \rangle = \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.28a}$$

$$\langle u^\star - u^j, x^j - x^\star \rangle = -B_f(x^j, x^\star; u^\star) - B_f(x^\star, x^j; u^j) \tag{B.28b}$$

Step 3, from (B.25) and (B.24) we know that

$$0 \leqslant B_f(x^\star, x^j; u^j) \tag{B.29a}$$

$$\langle u^j, x^j - x^{j+1} \rangle \leqslant \frac{\alpha^j}{2}\left\|u^j\right\|_2^2 + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant \frac{\gamma^2 \alpha^j}{2} + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.29b}$$

Summing up (B.27), (B.28a), $\alpha^j \times$(B.28b), $\alpha^j \times$(B.29a) and $\alpha^j \times$(B.29b), we obtain the following

$$\alpha^j (f(x^j) - f(x^\star)) \leqslant \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 + \frac{(\alpha^j \gamma)^2}{2} \tag{B.30}$$

where we also used the fact that $B_f(x^j, x^\star; u^\star) = f(x^j) - f(x^\star) - \langle u^\star, x^j - x^\star \rangle$. Summing up (B.30) from $j = 1$ to $j = k$ we obtain

$$\begin{aligned}\sum_{j=1}^{k} \alpha^j (f(x^j) - f(x^\star)) &\leqslant \frac{1}{2} \left\| x^1 - x^\star \right\|_2^2 - \frac{1}{2} \left\| x^{k+1} - x^\star \right\|_2^2 + \sum_{j=1}^{k} \frac{(\alpha^j \gamma)^2}{2} \\ &\leqslant \frac{1}{2} \left\| x^1 - x^\star \right\|_2^2 + \sum_{j=1}^{k} \frac{(\alpha^j \gamma)^2}{2}.\end{aligned} \tag{B.31}$$

Step 4, since f is convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. □

The above theorem shows that if we choose a sequence of α^k such that

$$\lim_{k \to \infty} \sum_{j=1}^{k} \alpha^j = \infty, \ \ \lim_{k \to \infty} \sum_{j=1}^{k} (\alpha^j)^2 < \infty,$$

then algorithm (B.24) converges asymptotically. If the max number of iteration k is fixed in advance as a computational budget, then one can simply let $\alpha^j \propto \frac{1}{\sqrt{k}}$, and the above theorem shows a convergence rate of $O(1/\sqrt{k})$.

B.2.2 Strongly convex case

We now consider the case where $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex and γ-Lipschitz. Using the notion of Bregman divergence, this is to say that

$$\frac{\mu}{2} \left\| x' - x \right\|_2^2 \leqslant B_f(x', x; u), \ \ \forall x, x' \in \mathbb{R}^n, \ \ u \in \partial f(x) \tag{B.32a}$$

$$\left\| u \right\|_2 \leqslant \gamma, \ \ \forall u \in \partial f(x), x \in \mathbb{R}^n \tag{B.32b}$$

With these additional assumption, the convergence property of (B.24) is as follows.

Theorem B.4. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, and* (B.32) *holds. Let sequence $\{x^j\}$ be generated by* (B.24) *with $\alpha^k = \frac{2}{\mu(k+1)}$, $x^\star$ be an optimizer of optimization*

(B.1), *and* $\overline{x}^k = \frac{2}{k(k+1)}\sum_{j=1}^{k} jx^j$. *Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{2\gamma^2}{\mu(k+1)}$$

Proof. Step 1, applying Lemma A.3 to (B.24) gives

$$0 \leqslant \langle x^{j+1} - (x^j - \alpha^j u^j), x^\star - x^{j+1}\rangle \tag{B.33}$$

Adding $\alpha^j\langle u^\star, x^j - x^\star\rangle$ to the both sides of the above inequality gives

$$\alpha^j\langle u^\star, x^j - x^\star\rangle \leqslant \langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle + \alpha^j\langle u^\star - u^j, x^j - x^\star\rangle + \alpha^j\langle u^j, x^j - x^{j+1}\rangle \tag{B.34}$$

Step 2, one can verify the following using (B.5) and (B.8)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle = \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.35a}$$

$$\langle u^\star - u^j, x^j - x^\star\rangle = -B_f(x^j, x^\star; u^\star) - B_f(x^\star, x^j; u^j) \tag{B.35b}$$

Step 3, using (B.32) and completing square we can show

$$\frac{\mu}{2}\left\|x^j - x^\star\right\|_2^2 \leqslant B_f(x^\star, x^j; u^j) \tag{B.36a}$$

$$\langle u^j, x^j - x^{j+1}\rangle \leqslant \frac{\alpha^j}{2}\left\|u^j\right\|_2^2 + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant \frac{\gamma^2\alpha^j}{2} + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.36b}$$

Summing up $\frac{1}{\alpha^j}\times$(B.34), $\frac{1}{\alpha^j}\times$(B.35a), (B.35b), (B.36a) and (B.36b), we obtain the following

$$f(x^j) - f(x^\star) \leqslant \left(\frac{1}{2\alpha^j} - \frac{\mu}{2}\right)\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2\alpha^j}\left\|x^{j+1} - x^\star\right\|_2^2 + \frac{\gamma^2\alpha^j}{2}$$

where we also used the fact that $B_f(x^j, x^\star; u^\star) = f(x^j) - f(x^\star) - \langle u^\star, x^j - x^\star\rangle$. Letting

$\alpha^j = \frac{2}{\mu(j+1)}$, then multiple the above inequality with j we have

$$\begin{aligned} j(f(x^j) - f(x^\star)) \leqslant& \frac{j(j-1)\mu}{4} \left\|x^j - x^\star\right\|_2^2 - \frac{j(j+1)\mu}{4} \left\|x^{j+1} - x^\star\right\|_2^2 + \frac{\gamma^2 j}{(j+1)\mu} \\ \leqslant& \frac{j(j-1)\mu}{4} \left\|x^j - x^\star\right\|_2^2 - \frac{j(j+1)\mu}{4} \left\|x^{j+1} - x^\star\right\|_2^2 + \frac{\gamma^2}{\mu} \end{aligned} \tag{B.37}$$

Summing up (B.37) from $j = 1$ to $j = k$ we obtain

$$\sum_{j=1}^{k} j(f(x^j) - f(x^\star)) \leqslant -\frac{k(k+1)\mu}{4} \left\|x^{j+1} - x^\star\right\|_2^2 + \frac{k\gamma^2}{\mu} \leqslant \frac{k\gamma^2}{\mu} \tag{B.38}$$

Step 4, since f is convex, $B_f(x, x^\star)$ is a convex function of x. Applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results.

□

B.3 Proximal gradient method

A popular paradigm in modern convex optimization is the following *composite objective optimization*

$$\underset{x}{\text{minimize}} \quad f(x) + h(x) \tag{B.39}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex, and $g : \mathbb{R}^n \to \mathbb{R}$ is a simple non-smooth convex function, then a popular variation of gradient descent method is the *proximal gradient method*, which iterates as follows

$$x^{k+1} = \underset{x}{\text{argmin}} \; \alpha\langle \nabla f(x^k), x\rangle + \alpha h(x) + \frac{1}{2} \left\|x - x^k\right\|_2^2 \tag{B.40}$$

Let $x^\star$ be an optimizer of optimization (B.39). Using Theorem A.2 (where $\mathbb{X} = \mathbb{R}^n$) to the optimization in (B.39) and (B.40) we have

$$0 = \nabla f(x^\star) + u^\star, \;\; u^\star \in \partial h(x^\star) \tag{B.41a}$$

$$0 =\alpha(\nabla f(x^k) + u^{k+1}) + x^{k+1} - x^k, \;\; u^{k+1} \in \partial h(x^{k+1}) \tag{B.41b}$$

In the remainder of this section, we will discuss the convergence property of the algorithm in (B.40) under different assumptions.

B.3.1 Non-strongly convex case

We start with the case where $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth and convex, but not necessarily strongly convex. Using the notion of Bregman divergence, this is to say that

$$0 \leqslant B_f(x', x) \leqslant \frac{\beta}{2} \|x' - x\|_2^2, \;\; \forall x, x' \in \mathbb{R}^n \tag{B.42}$$

With these additional assumption, the convergence property of (B.40) is as follows.

Theorem B.5. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, continuously differentiable, and* (B.42) *holds. Suppose function $g : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper. Let sequence $\{x^j\}$ be generated by* (B.40) *with $\alpha \leqslant \frac{1}{\beta}$, $x^\star$ be an optimizer of optimization* (B.39), *and $\overline{x}^k = \frac{1}{k}\sum_{j=1}^k x^{j+1}$. Then*

$$f(\overline{x}^k) + h(\overline{x}^k) - f(x^\star) - h(x^\star) \leqslant \frac{\|x^1 - x^\star\|_2^2}{2\alpha k}$$

Proof. Step 1, the inner product of (B.41b) with $x^\star - x^{j+1}$ is given by

$$0 = \langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle + \alpha\langle \nabla f(x^j), x^\star - x^{j+1}\rangle + \alpha\langle u^{k+1}, x^\star - x^{j+1}\rangle$$

Adding $\alpha\langle \nabla f(x^\star) + u^\star, x^{j+1} - x^\star\rangle$ with $u^\star \in \partial h(x^\star)$ to the both sides of the above equation gives

$$\begin{aligned}&\alpha\langle \nabla f(x^\star) + u^\star, x^{j+1} - x^\star\rangle \\ =&\langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle + \alpha\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star\rangle + \alpha\langle u^\star - u^{j+1}, x^{j+1} - x^\star\rangle\end{aligned} \tag{B.43}$$

Step 2, one can verify the following using (B.5) and (B.8)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle = \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.44a}$$

$$\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star\rangle = B_f(x^{j+1}, x^j) - B_f(x^{j+1}, x^\star) - B_f(x^\star, x^j) \tag{B.44b}$$

$$\langle u^\star - u^{j+1}, x^{j+1} - x^\star\rangle = -B_h(x^{j+1}, x^\star; u^\star) - B_h(x^\star, x^{j+1}; u^{j+1}) \tag{B.44c}$$

Step 3, since f is β-smooth and convex, g is convex, we know that

$$B_f(x^{j+1}, x^j) \leqslant \frac{\beta}{2}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.45a}$$

$$0 \leqslant B_f(x^\star, x^j) \tag{B.45b}$$

$$0 \leqslant B_h(x^\star, x^{j+1}; u^{j+1}) \tag{B.45c}$$

Summing up (B.43), (B.44a), $\alpha\times$(B.44b),$\alpha\times$(B.44c), $\alpha\times$(B.45a) and $\alpha\times$(B.45b), $\alpha\times$(B.45c), and use the assumption that $\alpha \leqslant \frac{1}{\beta}$, we obtain the following

$$\alpha(f(x^{j+1}) + h(x^{j+1}) - f(x^\star) - h(x^\star)) \leqslant \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 \tag{B.46}$$

where we also used the fact that $B_f(x^{j+1}, x^\star) = f(x^{j+1}) - f(x^\star) - \langle \nabla f(x^\star), x^{j+1} - x^\star\rangle$ and $B_h(x^{j+1}, x^\star; u^\star) = h(x^{j+1}) - h(x^\star) - \langle u^\star, x^{j+1} - x^\star\rangle$. Summing up (B.46) from $j = 1$ to $j = k$ we obtain

$$\alpha\sum_{j=1}^{k}(f(x^{j+1}) + h(x^{j+1}) - f(x^\star) - h(x^\star)) \leqslant \frac{1}{2}\left\|x^1 - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 \leqslant \frac{1}{2}\left\|x^1 - x^\star\right\|_2^2 \tag{B.47}$$

Step 4, since f and h are convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. □

B.3.2 Strongly convex case

What happens if we strengthen the assumptions in (B.42) by assuming $f : \mathbb{R}^n \to \mathbb{R}$ to be both smooth and strongly convex? *i.e.*,

$$\frac{\mu}{2}\left\|x' - x\right\|_2^2 \leqslant B_f(x', x) \leqslant \frac{\beta}{2}\left\|x' - x\right\|_2^2, \ \ \forall x, x' \in \mathbb{R}^n \tag{B.48}$$

The following theorem shows that, with the above assumptions, algorithm (B.40) ensures the quadratic distance to optimum shrinks exponentially.

Theorem B.6. *Suppose function* $f : \mathbb{R}^n \to \mathbb{R}$ *is closed, convex, proper, continuously differentiable, and* (B.48) *holds. Suppose function* $g : \mathbb{R}^n \to \mathbb{R}$ *is closed, convex, proper. Let sequence* $\{x^j\}$ *be generated by* (B.40) *with* $\alpha = \frac{1}{\beta}$*, and* $x^\star$ *be an optimizer of optimization* (B.1)*. Then*

$$\left\|x^{k+1} - x^\star\right\|_2^2 \leqslant \left(\frac{\beta-\mu}{\beta+\mu}\right)^k \left\|x^1 - x^\star\right\|_2^2$$

Proof. Step 1, summing up $-\alpha\times$(B.41a) and (B.41b), gives

$$0 = x^{j+1} - x^j + \alpha(\nabla f(x^j) - \nabla f(x^\star)) + \alpha(u^{j+1} - u^\star)$$

The inner product of the above equation with $x^\star - x^j$ is given by

$$0 = \langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle + \alpha\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star\rangle + \alpha\langle u^\star - u^{j+1}, x^{j+1} - x^\star\rangle \tag{B.49}$$

Step 2, one can verify the following using (B.5) and (B.8)

$$\langle x^{j+1} - x^j, x^\star - x^{j+1}\rangle = \frac{1}{2}\left\|x^j - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^\star\right\|_2^2 - \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.50a}$$

$$\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star\rangle = B_f(x^{j+1}, x^j) - B_f(x^{j+1}, x^\star) - B_f(x^\star, x^j) \tag{B.50b}$$

$$\langle u^\star - u^{j+1}, x^{j+1} - x^\star\rangle = -B_h(x^{j+1}, x^\star; u^\star) - B_h(x^\star, x^{j+1}; u^{j+1}) \tag{B.50c}$$

Step 3, since f is β-smooth and μ-strongly convex, we know that

$$B_f(x^{j+1}, x^j) \leqslant \frac{\beta}{2} \left\| x^{j+1} - x^j \right\|_2^2 \tag{B.51a}$$

$$\frac{\mu}{2} \left\| x^j - x^\star \right\|_2^2 \leqslant B_f(x^\star, x^j) \tag{B.51b}$$

$$\frac{\mu}{2} \left\| x^{j+1} - x^\star \right\|_2^2 \leqslant B_f(x^{j+1}, x^\star) \tag{B.51c}$$

$$0 \leqslant B_h(x^{j+1}, x^\star; u^\star) \tag{B.51d}$$

$$0 \leqslant B_h(x^\star, x^{j+1}; u^{j+1}) \tag{B.51e}$$

Summing up (B.49), (B.50a), $\alpha\times$(B.50b), $\alpha\times$(B.50c), $\alpha\times$(B.51a), $\alpha\times$(B.51b), $\alpha\times$(B.51c), $\alpha\times$(B.51d), and $\alpha\times$(B.51e), we obtain the following

$$\frac{1 + \mu/\beta}{2} \left\| x^{j+1} - x^\star \right\|_2^2 \leqslant \frac{1 - \mu/\beta}{2} \left\| x^j - x^\star \right\|_2^2 \tag{B.52}$$

where we also use the assumption that $\alpha = \frac{1}{\beta}$. Using the above inequality recursively for k times, we obtain the desired results. $\square$

B.4 Mirror descent method

In this section, we will discuss mirror descent method, an extension to projected gradient method in B.1 and projected subgradient method in B.2. Key to our discussion is the notion of distance generating function follows. Given constrained optimization (B.1), we say function $\psi : \mathbb{X} \to \mathbb{R}$ is a distance generating function if

- $\psi(x) = \infty$ for any $x \notin \mathbb{X}$, the gradient map $\nabla\psi : \mathbb{X} \to \mathbb{R}^n$ is well defined over $\mathbb{X}$.
- $\psi : \mathbb{X} \to \mathbb{R}$ is closed, convex, proper.
- ψ is 1-strongly convex, *i.e.*,

$$B_\psi(x', x) \geqslant \frac{1}{2} \left\| x' - x \right\|_2^2, \;\; \forall x, x' \in \mathbb{X} \tag{B.53}$$

B.4.1 Smooth case

A natural extension to the projected gradient method in B.1 is the following mirror descent method

$$x^{k+1} = \operatorname*{argmin}_{x \in \mathbb{X}} \; \alpha \langle \nabla f(x^k), x \rangle + B_\psi(x, x^k) \tag{B.54}$$

We assume $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth and convex, but not necessarily strongly convex, *i.e.*, the following conditions hold

$$0 \leqslant B_f(x', x) \leqslant \frac{\beta}{2} \|x' - x\|_2^2, \;\; \forall x, x' \in \mathbb{R}^n \tag{B.55}$$

With these assumptions, the convergence property of (B.54) is given as follows.

Theorem B.7. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, continuously differentiable, and* (B.55) *holds. Let sequence $\{x^j\}$ be generated by* (B.54) *with $\alpha \leqslant \frac{1}{\beta}$, $x^\star$ be an optimizer of optimization* (B.1)*, $\psi : \mathbb{X} \to \mathbb{R}$ be a distance generating function of optimization* (B.1)*, and $\overline{x}^k = \frac{1}{k}\sum_{j=1}^k x^{j+1}$. Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{B_\psi(x^\star, x^1)}{\alpha k}$$

Proof. Step 1, applying Theorem A.2 to (B.54) gives

$$0 \leqslant \langle \alpha \nabla f(x^j) + \nabla \psi(x^{j+1}) - \nabla \psi(x^j), x^\star - x^{j+1} \rangle \tag{B.56}$$

Adding $\alpha \langle \nabla f(x^\star), x^{j+1} - x^\star \rangle$ to the both sides of the above inequality gives

$$\alpha \langle \nabla f(x^\star), x^{j+1} - x^\star \rangle \leqslant \langle \nabla \psi(x^{j+1}) - \nabla \psi(x^j), x^\star - x^{j+1} \rangle + \alpha \langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle \tag{B.57}$$

Step 2, one can verify the following using (B.5)

$$\langle \nabla \psi(x^{j+1}) - \nabla \psi(x^j), x^\star - x^{j+1} \rangle = B_\psi(x^\star, x^j) - B_\psi(x^\star, x^{j+1}) - B_\psi(x^{j+1}, x^j) \tag{B.58a}$$

$$\langle \nabla f(x^\star) - \nabla f(x^j), x^{j+1} - x^\star \rangle = B_f(x^{j+1}, x^j) - B_f(x^{j+1}, x^\star) - B_f(x^\star, x^j) \tag{B.58b}$$

Step 3, since f is β-smooth and convex and ψ is 1-strongly convex, we know that

$$B_f(x^{j+1}, x^j) \leqslant \frac{\beta}{2} \left\| x^{j+1} - x^j \right\|_2^2 \tag{B.59a}$$

$$0 \leqslant B_f(x^\star, x^j) \tag{B.59b}$$

$$\frac{1}{2} \left\| x^{j+1} - x^j \right\|_2^2 \leqslant B_\psi(x^{j+1}, x^j) \tag{B.59c}$$

Summing up (B.57), (B.58a), $\alpha\times$(B.58b), $\alpha\times$(B.59a), $\alpha\times$(B.59b), and (B.59c), then using the assumption that $\alpha \leqslant \frac{1}{\beta}$, we obtain the following

$$\alpha(f(x^{j+1}) - f(x^\star)) \leqslant B_\psi(x^\star, x^j) - B_\psi(x^\star, x^{j+1}) \tag{B.60}$$

where we used the fact that $B_f(x^{j+1}, x^\star) = f(x^{j+1}) - f(x^\star) - \langle \nabla f(x^\star), x^{j+1} - x^\star \rangle$. Summing up (B.60) from $j = 1$ to $j = k$ we obtain

$$\alpha \sum_{j=1}^{k} (f(x^{j+1}) - f(x^\star)) \leqslant B_\psi(x^\star, x^1) - B_\psi(x^\star, x^{k+1}) \leqslant B_\psi(x^\star, x^1) \tag{B.61}$$

Step 4, since f is convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. □

B.4.2 Non-smooth case

When function f in (B.1) is not differentiable, a natural extension to (B.54) is the following algorithm

$$x^{k+1} = \underset{x \in \mathbb{X}}{\operatorname{argmin}} \ \alpha^k \langle u^k, x \rangle + B_\psi(x, x^k), \ \ u^k \in \partial f(x^k) \tag{B.62}$$

We will show that the convergence properties of the above algorithm closely mimic those of projected subgradient method in B.2.

B.4.2.1 Non-strongly convex case

We start with the case where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and γ-Lipschitz, *i.e.*, the following conditions hold.

$$0 \leqslant B_f(x', x; u), \ \ \forall x, x' \in \mathbb{R}^n, \ \ u \in \partial f(x) \tag{B.63a}$$

$$\|u\|_2 \leqslant \gamma, \ \ \forall u \in \partial f(x), x \in \mathbb{R}^n \tag{B.63b}$$

Theorem B.8. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, and* (B.63) *holds. Let sequence $\{x^j\}$ be generated by* (B.62)*, $x^\star$ be an optimizer of optimization* (B.1)*, $\psi : \mathbb{X} \to \mathbb{R}$ be a distance generating function of optimization* (B.1)*, and $\overline{x}^k = \frac{1}{\sum_{j=1}^k \alpha^j} \sum_{j=1}^k \alpha^j x^j$. Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{B_\psi(x^\star, x^1) + \frac{1}{2}\sum_{j=1}^k \gamma^2 (\alpha^j)^2}{\sum_{j=1}^k \alpha^j}$$

Proof. Step 1, applying Theorem A.2 to (B.62) gives

$$0 \leqslant \langle \alpha^j u^j + \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle \tag{B.64}$$

Adding $\alpha^j \langle u^\star, x^j - x^\star \rangle$ to the both sides of the above inequality gives

$$\alpha^j \langle u^\star, x^j - x^\star \rangle \leqslant \langle \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle + \alpha^j \langle u^\star - u^j, x^j - x^\star \rangle + \alpha^j \langle u^j, x^j - x^{j+1} \rangle \tag{B.65}$$

Step 2, one can verify the following using (B.5)

$$\langle \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle = B_\psi(x^\star, x^j) - B_\psi(x^\star, x^{j+1}) - B_\psi(x^{j+1}, x^j) \tag{B.66a}$$

$$\langle u^\star - u^j, x^j - x^\star \rangle = - B_f(x^j, x^\star; u^\star) - B_f(x^\star, x^j; u^j) \tag{B.66b}$$

Step 3, using (B.63) and the fact that ψ is 1-strongly convex, we can show

$$\begin{aligned} 0 \leqslant & B_f(x^\star, x^j; u^j) && \text{(B.67a)} \\ \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant & B_\psi(x^{j+1}, x^j) && \text{(B.67b)} \end{aligned}$$

In addition, by completing the square and using (B.63) one can show the following

$$\langle u^j, x^j - x^{j+1}\rangle \leqslant \frac{\alpha^j}{2}\left\|u^j\right\|_2^2 + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant \frac{\gamma^2\alpha^j}{2} + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.68}$$

Summing up (B.65), (B.66a), $\alpha^j\times$(B.66b), $\alpha^j\times$(B.67a), (B.67b), $\alpha^j\times$(B.68), we obtain the following

$$\alpha^j(f(x^j) - f(x^\star)) \leqslant B_\psi(x^\star, x^j) - B_\psi(x^\star, x^{j+1}) + \frac{(\alpha^j\gamma)^2}{2} \tag{B.69}$$

where we also used the fact that $B_f(x^j, x^\star; u^\star) = f(x^j) - f(x^\star) - \langle u^\star, x^j - x^\star\rangle$. Summing up (B.69) from $j = 1$ to $j = k$ we obtain

$$\begin{aligned} \sum_{j=1}^{k}\alpha^j(f(x^j) - f(x^\star)) \leqslant & B_\psi(x^\star, x^1) - B_\psi(x^\star, x^{k+1}) + \sum_{j=1}^{k}\frac{(\alpha^j\gamma)^2}{2} \\ \leqslant & B_\psi(x^\star, x^1) + \sum_{j=1}^{k}\frac{(\alpha^j\gamma)^2}{2} \end{aligned} \tag{B.70}$$

Step 4, since f is convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. $\square$

The above theorem shows that if we choose a sequence of α^k such that

$$\lim_{k\to\infty}\sum_{j=1}^{k}\alpha^j = \infty, \quad \lim_{k\to\infty}\sum_{j=1}^{k}(\alpha^j)^2 < \infty,$$

then algorithm (B.62) converges asymptotically. If the max number of iteration k is fixed in advance as a computational budget, then one can simply let $\alpha^j \propto \frac{1}{\sqrt{k}}$, and the above theorem shows a convergence rate of $O(1/\sqrt{k})$.

B.4.2.2 Strongly convex case

We now consider the case where $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex and γ-Lipschitz. Using the notion of Bregman divergence, this is to say that

$$\mu B_\psi(x', x) \leqslant B_f(x', x; u), \;\; \forall x, x' \in \mathbb{R}^n, \;\; u \in \partial f(x) \tag{B.71a}$$

$$\|u\|_2 \leqslant \gamma, \;\; \forall u \in \partial f(x), x \in \mathbb{R}^n \tag{B.71b}$$

With these additional assumption, the convergence property of (B.62) is as follows.

Theorem B.9. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, and* (B.71) *holds. Let sequence $\{x^j\}$ be generated by* (B.62) *with $\alpha^k = \frac{2}{\mu(k+1)}$, $x^\star$ be an optimizer of optimization* (B.1)*, $\psi : \mathbb{X} \to \mathbb{R}$ be a distance generating function of optimization* (B.1)*, and $\overline{x}^k = \frac{2}{k(k+1)} \sum_{j=1}^k j x^j$. Then*

$$f(\overline{x}^k) - f(x^\star) \leqslant \frac{2\gamma^2}{\mu(k+1)}$$

Proof. Step 1, applying Theorem A.2 to (B.62) and letting $x = x^{j+1}$ in (B.3) gives

$$0 \leqslant \langle \alpha^j u^j + \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle \tag{B.72}$$

Adding $\alpha^j \langle u^\star, x^j - x^\star \rangle$ to the both sides of the above inequality

$$\alpha^j \langle u^\star, x^j - x^\star \rangle \leqslant \langle \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle + \alpha^j \langle u^\star - u^j, x^j - x^\star \rangle + \alpha^j \langle u^j, x^j - x^{j+1} \rangle \tag{B.73}$$

Step 2, one can verify the following using (B.5)

$$\langle \nabla\psi(x^{j+1}) - \nabla\psi(x^j), x^\star - x^{j+1} \rangle = B_\psi(x^\star, x^j) - B_\psi(x^\star, x^{j+1}) - B_\psi(x^{j+1}, x^j) \tag{B.74a}$$

$$\langle u^\star - u^j, x^j - x^\star \rangle = - B_f(x^j, x^\star; u^\star) - B_f(x^\star, x^j; u^j) \tag{B.74b}$$

Step 3, using (B.71) and the fact that ψ is 1-strongly convex, we can show

$$\begin{aligned} \mu B_\psi(x^\star, x^j) \leqslant& B_f(x^\star, x^j; u^j) && \text{(B.75a)} \\ \frac{1}{2}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant& B_\psi(x^{j+1}, x^j) && \text{(B.75b)} \end{aligned}$$

In addition, by completing the square and using (B.71) one can show the following

$$\langle u^j, x^j - x^{j+1} \rangle \leqslant \frac{\alpha^j}{2}\left\|u^j\right\|_2^2 + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \leqslant \frac{\gamma^2\alpha^j}{2} + \frac{1}{2\alpha^j}\left\|x^{j+1} - x^j\right\|_2^2 \tag{B.76}$$

Summing up $\frac{1}{\alpha^j}\times$(B.73), $\frac{1}{\alpha^j}\times$(B.74a),(B.74b), (B.75a), $\frac{1}{\alpha^j}\times$(B.75b) and (B.76), we obtain the following

$$f(x^j) - f(x^\star) \leqslant \left(\frac{1}{\alpha^j} - \mu\right) B_\psi(x^\star, x^j) - \frac{1}{\alpha^j} B_\psi(x^\star, x^{j+1}) + \frac{\gamma^2\alpha^j}{2}$$

where we also used the fact that $B_f(x^j, x^\star; u^\star) = f(x^j) - f(x^\star) - \langle u^\star, x^j - x^\star \rangle$. Letting $\alpha^j = \frac{2}{\mu(j+1)}$, then multiplying the above inequality with j we have

$$\begin{aligned} j(f(x^j) - f(x^\star)) \leqslant& \frac{j(j-1)\mu}{2} B_\psi(x^\star, x^j) - \frac{j(j+1)\mu}{2} B_\psi(x^\star, x^{j+1}) + \frac{\gamma^2 j}{(j+1)\mu} \\ \leqslant& \frac{j(j-1)\mu}{2} B_\psi(x^\star, x^j) - \frac{j(j+1)\mu}{2} B_\psi(x^\star, x^{j+1}) + \frac{\gamma^2}{\mu} \end{aligned} \tag{B.77}$$

Summing up (B.77) from $j = 1$ to $j = k$ we obtain

$$\sum_{j=1}^{k} j(f(x^j) - f(x^\star)) \leqslant -\frac{k(k+1)\mu}{2} B_\psi(x^\star, x^j) + \frac{k\gamma^2}{\mu} \leqslant \frac{k\gamma^2}{\mu} \tag{B.78}$$

Step 4, since f is convex, applying the Jensen's inequality in (A.4) to the left hand side of the above inequality we obtain the desired results. $\square$

B.5 Conditional gradient method

In this section, we discuss a first order method designed for the optimization in (B.1) when

- f is convex and smooth
- $\mathbb{X}$ is closed, convex, and is contained within a ℓ_2-norm ball with finite radius

In other words, the following conditions hold

$$\langle \nabla f(x), x' - x \rangle \leqslant f(x') - f(x) \tag{B.79a}$$

$$f(x') - f(x) \leqslant \langle \nabla f(x), x' - x \rangle + \frac{\beta}{2} \|x' - x\|_2^2 \tag{B.79b}$$

$$\|x' - x\|_2 \leqslant \delta, \ \ \forall x', x \in \mathbb{X} \tag{B.79c}$$

The conditional gradient method, also known as the Frank-Wolfe method solves the optimization in (B.1) using the following iterations

$$y^k = \operatorname*{argmin}_{x \in \mathbb{X}} \ \langle \nabla f(x^k), x \rangle \tag{B.80a}$$

$$x^{k+1} = x^k + \alpha^k (y^k - x^k) \tag{B.80b}$$

Notice that (B.80a) ensures $y^k \in \mathbb{X}$ for all k. If $\alpha^k \in (0, 1)$, then $x^{k+1} = (1 - \alpha^k)x^k + \alpha^k y^k$ is a convex combination of points in convex set $\mathbb{X}$, hence itself is still in set $\mathbb{X}$.

Theorem B.10. *Suppose function* $f : \mathbb{R}^n \to \mathbb{R}$ *is closed, convex, proper, continuously differentiable, and* (B.79) *holds. Let sequence* $\{x^k\}$ *be generated by* (B.80) *with* $\alpha = \frac{2}{k+1}$*, and* $x^\star$ *be an optimizer of optimization* (B.1)*. Then*

$$f(x^k) - f(x^\star) \leqslant \frac{2\beta\delta^2}{k+1}$$

Proof. This proof uses a long chain of inequality that changes slightly at each step. For clarity, we underline the terms different from the previous step, and annotate the result used at each step above the corresponding equality or inequality sign.

Observe that

$$
\begin{aligned}
& f(x^{k+1}) - f(x^\star) - (f(x^k) - f(x^\star)) \\
\overset{\text{(B.79b)}}{\leqslant} & \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{\beta}{2} \left\| x^{k+1} - x^k \right\|_2^2 \\
\overset{\text{(B.80b)}}{=} & \underline{\alpha^k} \langle \nabla f(x^k), \underline{y^k - x^k} \rangle + \underline{\frac{\beta(\alpha^k)^2}{2} \left\| y^k - x^k \right\|_2^2} \\
\overset{\text{(B.80a)}}{\leqslant} & \alpha^k \langle \nabla f(x^k), \underline{x^\star} - x^k \rangle + \frac{\beta(\alpha^k)^2}{2} \left\| y^k - x^k \right\|_2^2 \\
\overset{\text{(B.79a)}}{\leqslant} & \alpha^k (\underline{f(x^\star) - f(x^k)}) + \frac{\beta(\alpha^k)^2}{2} \left\| y^k - x^k \right\|_2^2 \\
\overset{\text{(B.79c)}}{\leqslant} & \alpha^k (f(x^\star) - f(x^k)) + \frac{\beta(\alpha^k)^2}{2} \underline{\delta^2}
\end{aligned}
\tag{B.81}
$$

Hence

$$
f(x^{k+1}) - f(x^\star) \leqslant (1 - \alpha^k)(f(x^k) - f(x^\star)) + \frac{\beta \delta^2 (\alpha^k)^2}{2} \tag{B.82}
$$

Let $\alpha^k = \frac{2}{k+1}$, we prove the desired results using the following induction.

- if $k = 1$, $\alpha^1 = 1$, and we have

$$
f(x^2) - f(x^\star) \leqslant \frac{\beta \delta^2}{2} = \frac{\beta \delta^2}{k+1} \tag{B.83}
$$

- assume $f(x^k) - f(x^\star) \leqslant \frac{2\beta\delta^2}{k+1}$ for $k \geqslant 1$, then

$$
\begin{aligned}
f(x^{k+1}) - f(x^\star) \leqslant & (1 - \alpha^k)(f(x^k) - f(x^\star)) + \frac{\beta \delta^2 (\alpha^k)^2}{2} \\
\leqslant & \frac{2(k-1)\beta\delta^2}{(k+1)^2} + \frac{2\beta\delta^2}{(k+1)^2} \\
= & \frac{2k\beta\delta^2}{(k+1)^2} < \frac{2\beta\delta^2}{k+1}
\end{aligned}
\tag{B.84}
$$

which completes the proof.

$\square$

B.6 Accelerated mirror descent method

The accelerated gradient method, or Nestrove's method, has been one of the most popular method in modern convex optimization. In this section, we discuss a variant of Nestrov's method, named accelerated mirror descent method[2], which is due to Michael B. Cohen[3].

Similar to B.4, here we also use the notion of distance generating function $\psi : \mathbb{X} \to \mathbb{R}$ for optimization (B.1), which is assumed to be continuously differentiable and 1-strongly convex. We will also use its conjugate function given by

$$\psi^*(z) = \max_{x \in \mathbb{X}} \ \langle z, x \rangle - \psi(x), \ \ \forall z \in \mathbb{R}^n \tag{B.85}$$

and its conjugate gradient map given by

$$\nabla\psi^*(z) = \operatorname*{argmax}_{x \in \mathbb{X}} \ \langle z, x \rangle - \psi(x), \ \ \forall z \in \mathbb{R}^n \tag{B.86}$$

Using Fenchel-Young's inequality [152, Thm. 23.5], one can verify the following

$$B_{\psi^*}(z', z) = B_{\psi}(\nabla\psi^*(z), \nabla\psi^*(z')) \geqslant \frac{1}{2} \left\| \nabla\psi^*(z') - \nabla\psi^*(z) \right\|_2^2, \ \ \forall x, x' \in \mathbb{X} \tag{B.87}$$

where the last step is due to the 1-strong convexity of function ψ.

We will assume that function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, continuously differentiable, and β-smooth, *i.e.*, the following conditions hold

$$f(x) - f(x') \leqslant \langle \nabla f(x), x - x' \rangle \tag{B.88a}$$

$$f(x') - f(x) - \langle \nabla f(x), x' - x \rangle \leqslant \frac{\beta}{2} \left\| x' - x \right\|_2^2 \tag{B.88b}$$

---

[2]The author learned the materials in this section from an amazing lecture by Prof. Jelena Diakonikolas in 2019 ADSI summer school at University of Washington, Seattle. The author would like to thank all the organizers and participants of this event for their contributions.

[3]The paper was published by his co-authors after his tragic demise; see [37].

There are many different elegant interpretations of Nestrov's method. Here we adopt the techniques used in approximated duality gap theory, developed by Jelena Diakonikolas and Lorenzo Orrechia [45]. In the following, we will first construct a ordinary differential equation (ODE) along which a desired gap function converges to zero at a prescribed rate, then use careful dicretization scheme to obtain an algorithm whose discrete time behavior matches the continuous time model.

B.6.1 Continuous time algorithm

Given optimization (B.1), we wish to design a ODE for a continuous time trajectory $x(t)$: $[0, \infty) \to \mathbb{X}$ such that $f(x(t)) - f(x^\star)$ converges to zero at rate $O(1/\rho(t))$, where $x^\star$ is an optimizer to optimization (B.1), and $\rho(t) : [0, \infty) \to \mathbb{R}$ is an increasing scalar function. To this end, observe that (we use dot to denote the derivative with respect to t)

$$\begin{aligned}
&\frac{d}{dt}[\rho(t)(f(x(t)) - f(x^\star))] \\
=&\dot{\rho}(t)(f(x(t)) - f(x^\star)) + \rho(t)\langle \nabla f(x(t)), \dot{x}(t)\rangle \\
\overset{\text{(B.88a)}}{\leqslant}&\langle \nabla f(x(t)), \dot{\rho}(t)(x(t) - x^\star) + \rho(t)\dot{x}(t)\rangle \\
=&\langle \dot{\rho}(t)\nabla f(x(t)), x(t) + \frac{\rho(t)}{\dot{\rho}(t)}\dot{x}(t) - x^\star\rangle
\end{aligned} \tag{B.89}$$

Further, let $z^\star$ be such that $\nabla\psi(x^\star) = z^\star$, then

$$\frac{d}{dt}B_{\psi*}(z(t), z^\star) = \langle \nabla\psi^*(z(t)) - \nabla\psi^*(z^\star), \dot{z}(t)\rangle = \langle \nabla\psi^*(z(t)) - x^\star, \dot{z}(t)\rangle \tag{B.90}$$

Summing up (B.89) and (B.90) we have

$$\begin{aligned}
&\frac{d}{dt}(\rho(t)(f(x(t)) - f(x^\star)) + B_{\psi*}(z(t), z^\star)) \\
\leqslant&\langle x(t) + \frac{\rho(t)}{\dot{\rho}(t)}\dot{x}(t) - x^\star, \dot{\rho}(t)\nabla f(x(t))\rangle + \langle \nabla\psi^*(z(t)) - x^\star, \dot{z}(t)\rangle
\end{aligned} \tag{B.91}$$

Therefore, if we let

$$
\begin{aligned}
x(t) + \frac{\rho(t)}{\dot\rho(t)}\dot x(t) =& \nabla\psi^*(z(t)) \\
\dot z(t) =& -\dot\rho(t)\nabla f(x(t))
\end{aligned}
\tag{B.92}
$$

for all $t \in [0,\infty)$, we must have

$$
\frac{d}{dt}(\rho(t)(f(x(t)) - f(x^\star)) + B_{\psi*}(z(t), z^\star)) \leqslant 0 \tag{B.93}
$$

which then implies

$$
\begin{aligned}
f(x(t)) - f(x^\star)) \leqslant& \frac{\rho(t)(f(x(t)) - f(x^\star)) + B_{\psi*}(z(t), z^\star))}{\rho(t)} \\
\leqslant& \frac{\rho(0)(f(x(0)) - f(x^\star)) + B_{\psi*}(z(0), z^\star))}{\rho(t)}
\end{aligned}
\tag{B.94}
$$

In other words, $f(x(t)) - f(x^\star))$ converges to zero at the rate of $O(1/\rho(t))$. Rewriting (B.92) gives the following ODE

$$
\begin{aligned}
\frac{d}{dt}(\rho(t)x(t)) =& \dot\rho(t)\nabla\psi^*(z(t)) \\
\frac{d}{dt}z(t) =& -\dot{\rho(t)}\nabla f(x(t))
\end{aligned}
\tag{B.95}
$$

A special case of (B.95) was introduced in the seminal work by Su, Boyd and Candes [161]. An ODE equivalent to (B.95) was derived separately; we refer the interested readers to [95] and [180] for details.

B.6.2 Discrete time algorithm

Discretizing the above ODE using a predictor-corrector scheme we obtain the following algorithm

$$
\rho^{k+1} = \rho^k + \alpha^{k+1} \tag{B.96a}
$$

$$
x^{k+1} = \frac{\rho^k}{\rho^{k+1}}y^k + \frac{\alpha^{k+1}}{\rho^{k+1}}\nabla\psi^*(z^k) \tag{B.96b}
$$

$$z^{k+1} = z^k - \alpha^{k+1}\nabla f(x^{k+1}) \tag{B.96c}$$

$$y^{k+1} = \frac{\rho^k}{\rho^{k+1}}y^k + \frac{\alpha^{k+1}}{\rho^{k+1}}\nabla\psi^*(z^{k+1}) \tag{B.96d}$$

where $\rho^1 = \alpha^1$ and $\alpha^k > 0$ for all k. Intuitively, x^{k+1} and y^{k+1} in the above difference equation are different versions of the same variable. The value of x^{k+1} is computed first and used as the "predicted value" in computing z^{k+1}. Once z^{k+1} is computed, it is used to compute the "corrected value" given by y^{k+1}. From (B.86) we know that $\nabla\psi(z^{k+1}) \in \mathbb{X}$, and (B.96) ensures that y^k is a convex combination of y^k and $\nabla\psi^*(x^{k+1})$. Therefore as long as $y^1 \in \mathbb{X}$, we must have $y^k \in \mathbb{X}$ for all k.

The proofs we are about to introduce uses long chains of inequalities that changes slightly at each step. For clarity, we underline the terms different from the previous step, and annotate the result used at each step above the corresponding equality or inequality sign.

Theorem B.11. *Suppose function $f : \mathbb{R}^n \to \mathbb{R}$ is closed, convex, proper, continuously differentiable, and* (B.88) *holds. Let sequence $\{x^j\}$ be generated by* (B.96) *with $\alpha^k = \frac{k}{2\beta}$ and $\rho^1 = \alpha^1$, $x^\star$ be an optimizer of optimization* (B.1)*, $\psi : \mathbb{X} \to \mathbb{R}$ be a distance generating function of optimization* (B.1)*, and $z^\star = \nabla\psi(x^\star)$, $x^1 = y^1$, $z^1 = \nabla\psi(y^1)$. Then*

$$f(y^k) - f(x^\star) \leqslant \frac{2(f(y^1) - f(x^\star) + B_\psi(z^1, z^\star))}{k(k+1)},$$

Proof. First, observe that

$$
\begin{aligned}
& \rho^{k+1}(f(y^{k+1}) - f(x^\star)) - \rho^k(f(y^k) - f(x^\star)) \\
\overset{(B.96a)}{=} & \rho^{k+1}(f(y^{k+1}) - \underline{f(x^{k+1})}) + \rho^k(\underline{f(x^{k+1})} - f(y^k)) + \alpha^{k+1}(\underline{f(x^{k+1})} - f(x^\star)) \\
\overset{(B.88a)}{\leqslant} & \rho^{k+1}(f(y^{k+1}) - f(x^{k+1})) + \rho^k\underline{\langle \nabla f(x^{k+1}), x^{k+1} - y^k\rangle} + \alpha^{k+1}\underline{\langle \nabla f(x^{k+1}), x^{k+1} - x^\star\rangle} \\
\overset{(B.96a)}{=} & \rho^{k+1}(f(y^{k+1}) - f(x^{k+1})) + \langle \nabla f(x^{k+1}), \underline{\rho^{k+1}x^{k+1}} - \rho^k y^k - \alpha^{k+1}x^\star\rangle \\
= \; & \rho^{k+1}(f(y^{k+1}) - f(x^{k+1})) + \rho^{k+1}\langle \nabla f(x^{k+1}), x^{k+1} - \underline{y^{k+1}}\rangle \\
& + \langle \nabla f(x^{k+1}), \underline{\rho^{k+1}y^{k+1}} - \rho^k y^k - \alpha^{k+1}x^\star\rangle \\
\overset{(B.96d)}{=} & \rho^{k+1}(f(y^{k+1}) - f(x^{k+1}) - \langle \nabla f(x^{k+1}), y^{k+1} - x^{k+1}\rangle) \\
& + \underline{\alpha^{k+1}\langle \nabla f(x^{k+1}), \nabla\psi^*(z^{k+1}) - x^\star\rangle} \\
\overset{(B.88b)}{\leqslant} & \underline{\frac{\beta\rho^{k+1}}{2}\left\|y^{k+1} - x^{k+1}\right\|_2^2} + \alpha^{k+1}\langle \nabla f(x^{k+1}), \nabla\psi^*(z^{k+1}) - x^\star\rangle \\
\overset{(B.96)}{=} & \underline{\frac{\beta(\alpha^{k+1})^2}{2\rho^{k+1}}\left\|\nabla\psi^*(z^{k+1}) - \nabla\psi^*(z^k)\right\|_2^2} + \alpha^{k+1}\langle \nabla f(x^{k+1}), \nabla\psi^*(z^{k+1}) - x^\star\rangle
\end{aligned}
\tag{B.97}
$$

Next, observe the following

$$
\begin{aligned}
& B_{\psi^*}(z^{k+1}, z^\star) - B_{\psi^*}(z^k, z^\star) \\
= \; & - B_{\psi^*}(z^k, z^{k+1}) + \langle \nabla\psi^*(z^{k+1}) - \nabla\psi^*(z^\star), z^{k+1} - z^k\rangle \\
\overset{(B.87)}{\leqslant} & - \underline{\frac{1}{2}\left\|\nabla\psi^*(z^{k+1}) - \nabla\psi^*(z^k)\right\|^2} + \langle \nabla\psi^*(z^{k+1}) - x^\star, z^{k+1} - z^k\rangle \\
\overset{(B.96)}{=} & - \frac{1}{2}\left\|\nabla\psi^*(z^{k+1}) - \nabla\psi^*(z^k)\right\|^2 \underline{-\alpha^{k+1}\langle \nabla f(x^{k+1}), \nabla\psi^*(z^{k+1}) - x^\star\rangle}
\end{aligned}
\tag{B.98}
$$

Summing up (B.97) and (B.98) we obtain

$$
\begin{aligned}
& [\rho^{k+1}(f(y^{k+1}) - f(x^\star)) + B_{\psi^*}(z^{k+1}, z^\star)] - [\rho^k(f(y^k) - f(x^\star)) + B_{\psi^*}(z^k, z^\star)] \\
\leqslant & \frac{1}{2}\left(\frac{\beta(\alpha^{k+1})^2}{\rho^{k+1}} - 1\right)\left\|\nabla\psi^*(z^{k+1}) - \nabla\psi^*(z^k)\right\|^2
\end{aligned}
\tag{B.99}
$$

Letting $\alpha^k = \frac{k}{2\beta}$, then $\rho^k = \sum_{i=1}^k \alpha^k = \frac{k(k+1)}{4\beta}$ and

$$\frac{\beta(\alpha^{k+1})^2}{\rho^{k+1}} - 1 = \frac{k+1}{k+2} - 1 < 0,$$

Substituting the above inequality into (B.99) we have

$$\begin{aligned} &\rho^{k+1}(f(y^{k+1}) - f(x^\star)) + B_{\psi*}(z^{k+1}, z^\star) \\ \leqslant &\rho^k(f(y^k) - f(x^\star)) + B_{\psi*}(z^k, z^\star) \leqslant \cdots \leqslant \alpha^1(f(y^1) - f(x^\star)) + B_{\psi*}(z^1, z^\star) \end{aligned} \tag{B.100}$$

Since $B_{\psi*}(z^{k+1}, z^\star) \geqslant 0$, this implies

$$f(y^k) - f(x^\star) \leqslant \frac{\alpha^1(f(y^1) - f(x^\star)) + B_{\psi*}(z^1, z^\star)}{\rho^k} = \frac{2(f(y^1) - f(x^\star) + B_{\psi*}(z^1, z^\star))}{k(k+1)},$$

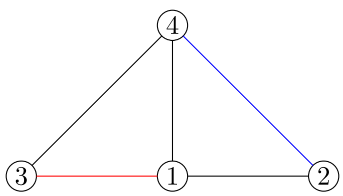
which completes the proof.

$\square$

C Graph and incidence matrices

We start with the definition of undirected graphs and its incidence matrix.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V} = \{1, 2, \ldots, |\mathcal{V}|\}$ and an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where an edge is a pair of distinct nodes in $\mathcal{V}$. We say graph $\mathcal{G}$ is connected if there exists a sequence of edges connecting any arbitrary two distinct nodes in $\mathcal{V}$. If we assign each edge in $\mathcal{E}$ with a direction, *i.e.*, a head and a tail, we obtain a directed graph, and the $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix is denoted by $D(\mathcal{G})$. Columns of $D(\mathcal{G})$ are indexed by the edges in $\mathcal{E}$, and the entry on their i-th row is "1" if node i is the head of the edge, "-1" if it is its tail, and 0 otherwise. If graph $\mathcal{G}$ is connected, the nullspace of $D(\mathcal{G})^\top$ is always spanned by $\mathbf{1}_{|\mathcal{V}|}$.



$$D(\mathcal{G}) = \left.\underbrace{\begin{bmatrix} 1 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & -1 & -1 \end{bmatrix}}_{|\mathcal{E}|}\right\} |\mathcal{V}|$$

Figure C.1: An example of undirected graph and its incidence matrix

D Linear systems

A discrete time linear time invariant system is described by the following difference equations

$$x_{t+1} = Ax_t + Bu_t \tag{D.1a}$$

$$y_t = Cx_t + Du_t \tag{D.1b}$$

where $u_t \in \mathbb{R}^m$, $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^p$ denote the input, state, and, respectively, output of the system at discrete time $k \in \mathbb{N}$.

D.1 Controllability & observability

The k-th order controllability matrix and observability matrix of system (D.1) are defined by

$$R_k = \begin{bmatrix} B & AB & \cdots & A^{k-1}B \end{bmatrix}, \quad O_k = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \tag{D.2}$$

We say system (D.1) is

- controllable if $\mathbf{rank}\, R_n = n$
- observable if $\mathbf{rank}\, O_n = n$

D.2 Ho-Kalman algorithm

The Ho-Kalman algorithm computes a realization of system (D.1) using its impulsive input-output data. In the following, we assume for simplicity that system (D.1) is both controllable and observable[1]. In particular, the algorithm uses the following Markov parameters of system (D.1)

$$M_k = \begin{cases} D, & k = 0 \\ CA^{k-1}B, & k = 1, 2, \dots \end{cases} \tag{D.3}$$

to compute system matrices in (D.1) up to a similarity transformation, that is,

$$(T^{-1}AT, T^{-1}B, CT, D) = (A_T, B_T, C_T, D_T) \tag{D.4}$$

where $T \in \mathbb{R}^{n \times n}$ is non-singular.

To see how the algorithm works, consider the following Hankel matrix

$$H_{n+1} = \begin{bmatrix} M_1 & M_2 & \dots & M_{n+1} \\ M_2 & M_3 & \dots & M_{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n+1} & M_{n+2} & \dots & M_{2n+1} \end{bmatrix} = \begin{bmatrix} CB & CAB & \dots & CA^nB \\ CAB & CA^2B & \dots & CA^{n+1}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^nB & CA^{n+1}B & \dots & CA^{2n}B \end{bmatrix} = O_{n+1}R_{n+1} \tag{D.5}$$

with its singular value decomposition given by

$$H_{n+1} = \begin{bmatrix} U_n & \bar{U}_n \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_n^\top \\ \bar{V}_n^\top \end{bmatrix} = U_n \Sigma_n V_n^\top \tag{D.6}$$

where $\Sigma_n \in \mathbb{R}^{n \times n}$ and $\mathbf{rank}\, \Sigma_n = n$ (since $H_{n+1} = O_{n+1}R_{n+1}$ has rank n). Let

$$T = R_{n+1} V_n \Sigma_n^{-\frac{1}{2}}. \tag{D.7}$$

[1] If system (D.1) is uncontrollable or unobservable (or both), the Ho-Kalman algorithm will compute a reduced order system realization that has the same input-output behavior as system (D.1)

Notice that T is non-singular since $\mathbf{rank}\, O_{n+1} = n$. Then

$$U_n \Sigma_n^{\frac{1}{2}} = O_{n+1} T = \begin{bmatrix} CT \\ CAT \\ \vdots \\ CA^{k-1}T \end{bmatrix} = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{k-1} \end{bmatrix} \tag{D.8}$$

and $T\Sigma_n^{\frac{1}{2}} V_n^\top = R_{n+1}$, which means

$$\Sigma_n^{\frac{1}{2}} V_n^\top = T^{-1} R_{n+1} = \begin{bmatrix} T^{-1}B & T^{-1}AB & \cdots & T^{-1}A^n B \end{bmatrix} = \begin{bmatrix} B_T & A_T B_T & \cdots & A_T^n B_T \end{bmatrix} \tag{D.9}$$

With these observations, we give the following Ho-Kalman algorithm that computes (A_T, B_T, C_T, D_T) based on the first $2n+1$ Markov parameters in (D.3)

1. construct matrix (D.5) and compute its SVD in (D.6)

2. B_T equals the first m columns of $\Sigma_n^{\frac{1}{2}} V_n^\top$, C_T equals the first p rows of $U_n \Sigma_n^{\frac{1}{2}}$, D_T equals M_0

3. A_T is the solution to the following linear equations system

$$U_n(1:(n-1)p, :) A_T = U_n(p+1:np, :)$$

where $U_n(i:j, :)$ equals the $i, i+1, \ldots, j$-th column of U_n