A Convex Optimisation Approach to Optimal Control in Queueing Systems

Víctor Valls

School of Computer Science and Statistics

Trinity College Dublin

Submitted for the Degree of Doctor of Philosophy

Supervised by Prof. Douglas Leith

2017

Abstract

Convex optimisation and max-weight are central topics in networking and control, and having a clear understanding of their relationship and what this involves is crucial from a theoretical and practical point of view. In this thesis we investigate how max-weight fits into convex optimisation from a pure convex approach without using fluid limits or Lyapunov optimisation. That is, we study how to equip convex optimisation with discrete actions and allow it to make optimal control decisions without previous knowledge of the mean packet arrival rate into the system.

Our results are sound and show that max-weight approaches can be encompassed within the body of convex optimisation. In particular, max-weight is a special case of the stochastic dual subgradient method with ϵ_k-subgradients and constant step size. We clarify the fundamental properties required for convergence, and bring to the fore the use of ϵ_k-subgradients as a key component for modelling problem characteristics apart from discrete actions. One of the great advantages of our approach is that optimal scheduling policies can be decoupled from the choice of convex optimisation algorithm or subgradient used to solve the dual problem, and as a result, it is possible to design scheduling policies with a high degree of flexibility. We illustrate the power of the analysis with three applications: the design of a traffic signal controller; distributed and asynchronous packet transmissions; and scheduling packets in networks where there are costs associated to selecting discrete actions.

The work in this thesis brings clarity and a fresh outlook to network optimisation problems, but also makes max-weight accessible from a convex optimisation perspective and therefore to a wider audience beyond networking and control. We believe that making max-weight accessible from a convex optimisation viewpoint will help to light the way to discovering new applications currently not covered by either max-weight approaches or convex optimisation.

Acknowledgements

A special thank you to my supervisor, Prof. Doug Leith, for his guidance and support during these four years. I feel fortunate to have had him as a supervisor; his vibrant approach to research has made a big impact on me on a professional and personal level. I would also like to thank Dr. Xavier Costa for giving me the opportunity to visit the NEC Laboratories in Heidelberg, and Prof. Boris Bellalta for his guidance during the early stages of my career.

Thanks also to all my friends in Ireland, Germany, and Spain. To my friends and colleagues at Trinity College and the Hamilton Institute: Cristina Cano, Alessandro Checco, Saman Feghhi, Andrés García-Saavedra, Naoise Holohan, Mohammad Karzand, Julien Monteil and Giulio Prevedello; to my friends in Mannheim, Tim Schneider and Vita Spanier, without whom I would have not had so much fun during my time in Germany; and to all my friends in Girona and Barcelona that are too many to list, but hope to see very soon.

Finally, I would like to thank my brother and parents for all their support and love.

My doctoral studies have been supported by Science Foundation Ireland under Grant No. 11/PI/1177.

Contents

CHAPTER 1

Introduction

Convexity plays a central role in mathematical optimisation from both a theoretical and a practical point of view. Some of the great advantages of formulating a problem as a convex optimisation is that there exist numerical methods that can solve the optimisation problem in a reliable and very efficient manner, and that a solution is always a global optimum. When an optimisation problem is not of the convex type, then one enters into the realm of non-convex optimisation where the specific structure of a problem must be exploited in order to obtain a solution, often not necessarily the optimal one. Nevertheless, there are some special cases of algorithms that can find optimal solutions to non-convex problems. One is the case of max-weight scheduling, an algorithm for scheduling packets in queueing networks which has received a lot of attention in the networking and control communities over the last years.

In short, max-weight was proposed by Tassiulas and Ephremides in their seminal paper [TE92]. It considers a network of interconnected queues in a slotted time system where packets arrive in each time slot and a network controller has to make a discrete (so non-convex) scheduling decision as to which packets to serve from each of the queues. Appealing features of max-weight are that the action set matches the actual decision variables (namely, do we transmit or not); that scheduling decisions are made without previous knowledge of the mean packet arrival rate in the system; and that it is able to stabilise the system (maximise the throughput) whenever there exists a policy or algorithm that would do so. These features have made max-weight well liked in the community and have fostered the design of extensions that consider (convex) utility functions, systems with time-varying channel capacity and connectivity, heavy-tailed traffic, *etc.* When dealing with utility functions, max-weight approaches usually start by formulating a fluid or convex optimisation, and then use the solution to that problem to design a scheduler for a discrete time system. One of the reasons for considering such an approach is that the fluid or convex formulation provides the big picture of the problem, whereas max-weight deals with the specific problem at a packet time scale [KY14, page 9]. Another reason used in the literature is that the dynamics of the Lagrange multipliers (associated to linear inequality constraints) in the dual subgradient method are like a queue update [SY13, Section 6.1]. However, the exact nature of the relationship between these two quantities remains still unclear, and so the body of work on max-weight approaches is separate from the mainstream literature on convex optimisation.

The connections between max-weight and convex optimisation are of much interest in the networking and control communities. There are several reasons for this. First, a better understanding of the structure of the problem. Encompassing max-weight within convex optimisation would allow us to characterise the solutions (optimality conditions) and to exploit this to improve the performance of queueing systems. An example of this is the work in [HN11], where the Lagrange multipliers of a convex optimisation are used to improve the delay of a max-weight scheduler. Second, it would allow the wealth of the well-established theory in convex optimisation to be used to bring features to max-weight that are currently not available. For example, asynchronous updates or packets transmissions. Third, clarity and simplicity. The popularity of max-weight has produced so many variants of the algorithm that the state of the art is becoming increasingly complex. There is a need for abstraction and to put concepts in a unified framework. Fourth, accessibility and dissemination. Convex optimisation is widely used in many fields, and making max-weight features available in convex optimisation would allow us to bring its benefits to other areas beyond networking and control. Fifth, a new outlook. On the way to finding the connections between max-weight and convex optimisation, we might shed light on properties in convex optimisation that might be otherwise obscured.

1.1. Purpose of the Thesis

Convex optimisation and max-weight are central topics in networking and control, and having a clear understanding of their relationship and what this involves is crucial. In this thesis we investigate how max-weight fits into convex optimisation from a pure convex approach and without invoking Foster-Lyapunov theorem. Namely, how to equip convex optimisation with discrete actions and allow it to make optimal decisions without knowledge of the mean packet arrival rate in the network. One of the objectives of this thesis is to provide a concise and comprehensive approach to max-weight in the terms used in the mainstream literature in convex optimisation [BV04, Ber99, BNO03]. That is, in the terms that somebody familiar with dual methods in convex optimisation should be easily able to understand. We believe that making max-weight accessible from a convex optimisation viewpoint will help to light the way to discovering new applications currently not covered by either max-weight approaches or convex optimisation.

1.2. Contributions

(i) *Generalising Max-Weight.* Our analysis places max-weight firmly within the field of convex optimisation, and rigorously shows it is a special case of the stochastic dual subgradient method with ϵ_k-subgradients and constant step size. This is shown in the optimisation framework presented in Chapter 3 in terms of elementary perturbations.

(ii) *Unifying Theoretical Framework.* In generalising max-weight and dual subgradient methods our analysis clarifies the fundamental properties required. In particular, that the objective function and constraints need only to be convex (probably non-differentiable); that the stochastic "noise" in the dual update must be ergodic and have finite variance; and that the ϵ_k perturbations on the multipliers must be bounded. Furthermore, our analysis brings to the fore ϵ_k-subgradients as a key ingredient for modelling discrete control problems, and capturing features such as asynchronous dual updates in an easy manner.

(iii) *Lagrange Multipliers, Approximate Lagrange Multipliers and Queues.* We establish a clean and clear connection between queues and Lagrange multipliers. In particular, α-scaled queue occupancies are quantities that stay close to the Lagrange multipliers in the dual subgradient method, and can be regarded as an approximation of the true Lagrange multipliers. In some special cases α-scaled queues are *exactly* Lagrange multipliers. This is first shown in Chapter 2, and later abstracted in the form of perturbations or ϵ_k-subgradients in Chapter 3.

(iv) *Decoupling Subgradients from Actions.* By using ϵ_k-subgradients and the continuity of the Skorokhod map, we show that it is possible to design scheduling policies (sequences of discrete actions) that are decoupled from a specific choice of (sub)gradient. This is of fundamental importance because it separates the design of scheduling policies from the characteristics of the optimisation algorithm or type of "descent" (*e.g.* gradient, subgradient, newton, proximal, *etc.*). Another great advantage of this approach is that it is possible to design scheduling policies that allow some flexibility on the order in which actions can be selected, which is key in systems that have constraints or penalties associated to selecting subsets of control actions.

(v) *Applications.* In Chapter 5 we provide a range of examples that show the power of the results in a concise and comprehensive manner. The examples provided include the design of a traffic signal controller; distributed and asynchronous packet transmissions; and scheduling of packets in networks where there are costs associated to selecting discrete actions.

This thesis has contributed to the literature with the following publications:

- V. Valls and D. J. Leith, "A convex optimization approach to discrete optimal control," in IEEE Transactions on Automatic Control (accepted).
- V. Valls and D. J. Leith, "Max-weight revisited: Sequences of nonconvex optimizations solving convex optimizations," in IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2676-2689, Oct. 2016.
- V. Valls and D. J. Leith, "Subgradient methods with perturbations in network problems," 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, 2016, pp. 482-487.
- V. Valls, J. Monteil and M. Bouroche, "A convex optimisation approach to traffic signal control," IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 2016, pp. 1508-1515.
- V. Valls and D. J. Leith, "Stochastic subgradient methods with approximate Lagrange multipliers," IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 6186-6191.
- V. Valls and D. J. Leith, "Dual subgradient methods using approximate multipliers," 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, 2015, pp. 1016-1021.
- V. Valls and D. J. Leith, "On the relationship between queues and multipliers," 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, 2014, pp. 250-255.

1.3. Notation

The notation used in this thesis is a blend of the notation used in standard reference books in convex optimisation [BV04, BNO03, Roc84].

The sets of natural, integers and real numbers are denoted by $\mathbf{N}$, $\mathbf{Z}$ and $\mathbf{R}$. We use $\mathbf{R}_+$ and $\mathbf{R}^n$ to denote the set of non-negative real numbers and n-dimensional real vectors. Similarly, we use $\mathbf{R}^{m\times n}$ to denote the set of $m \times n$ real matrices. Vectors and matrices are usually written, respectively, in lower and upper case, and all vectors are in column form. The transpose of a vector $x \in \mathbf{R}^n$ is indicated with x^T, and we use $\mathbf{1}$ to indicate the all ones vector. The Euclidean, ℓ_1 and ℓ_∞ norms of a vector $x \in \mathbf{R}^n$ are indicated, respectively, with $\|x\|_2$, $\|x\|_1$ and $\|x\|_\infty$.

Since we usually work with sequences we will use subscript to indicate an element in a sequence, and parenthesis to indicate an element in a vector. For example, for a sequence $\{x_k\}$ of vectors from $\mathbf{R}^n$ we have that

$$x_k = [x_k(1), \dots, x_k(n)]^T$$

where $x_k(j),\ j = 1, \dots, n$ is the j'th component of the k'th vector in the sequence. For two points $x, y \in \mathbf{R}^n$ we write $x \succ y$ when the $x(j) > y(j)$ for all $j = 1, \dots, n$, and $x \succeq y$ when $x(j) \geq y(j)$. For a constant $p > 0$ we will use operator $[\cdot]^{[0,p]}$ to denote the projection of a vector $x \in \mathbf{R}^n$ onto $[0, p]^n$, *i.e.*

$$[x]^{[0,p]} = [\max\{x(1), p\}, \dots, \max\{x(n), p\}]^T.$$

When $p = \infty$, to streamline notation we use $[\cdot]^+$ rather than $[\cdot]^{[0,\infty)}$.

The rest of the notation will be defined as needed in each of the chapters.

1.4. Related Work

Max-Weight. Max-weight scheduling was introduced by Tassiulas and Ephremides in their seminal paper [TE92]. They consider a network of queues with slotted time, an integer number of packet arrivals in each slot and a finite set of admissible scheduling patterns, referred to as *actions*. Using a Foster-Lyapunov approach, they present a scheduling policy that stabilises the queues in the system provided the mean packet arrival rate is strictly in the interior of the network capacity region. The scheduling policy consists of selecting the action at each slot that maximises the queue-length-weighted sum of rates, $y_k \in \arg\max_{y\in Y} -Q_k^T Ay$ where Y is the action set containing all the scheduling patterns, and A an incidence matrix representing the interconnection between the queues in the network.

Independently, [Sto05, ES05, NMR03] proposed extensions to the max-weight approach to accommodate concave utility functions. In [Sto05] the *greedy primal-dual* algorithm is introduced for network linear constraints and utility functions that are continuously differentiable and concave. The previous work is extended in [Sto06] to consider general nonlinear constraints. In [ES05] the utility fair allocation of throughput in a cellular downlink is considered, where the utility function used can be tuned to obtain different fairness criteria, but it is always twice differentiable. Queue departures are scheduled according to $x_k \in \arg\max_{x \in \text{conv}(Y)} -Q_k^T Ax$, and queue arrivals are scheduled by a congestion controller such that $\mathbf{E}(b_k(j)|Q_k) = \min\{\alpha_j K/(Q_k(j))^m, M\}$ and $\mathbf{E}((b_k(j))^2|Q_k) \leq B$ where α_j, K, m, B, M are positive constants. The work in [NMR03] considers power allocation in a multi-beam downlink satellite communication link with the aim of maximising throughput while ensuring queue stability. This is extended in a sequence of papers [NMR05, Nee06, NML08] and a book [Nee10b] to develop the *drift-plus-penalty* approach. In this approach, the basic strategy for scheduling queue departures is according to $y_k \in \arg\max_{y \in Y} -Q_k^T Ay$ and utility functions are incorporated in a variety of ways. For example, for concave non-decreasing continuous utility functions U of the form $U(x) = \sum_{j=1}^{n} U_j(x(j))$ one formulation is for a congestion controller to schedule arrivals into an ingress queue such that $b_k(j) \in \arg\max_{0 \leq b \leq R} VU_j(b) - bQ_k(j)$ where V, R are sufficiently large design parameters and $b \in \mathbf{R}$ [GNT06]. More advanced versions of the *drift-plus-penalty* algorithm make use of virtual queues to capture non-linear constraints, and allow the utility function and constraints to be non-differentiable. The recent work in [Nee14] presents a simplified convergence analysis of the *drift-plus-penalty*, and in [YN15] the algorithm is used to solve strongly convex programmes in a deterministic setting.

Max-weight is not always able to stabilise a system. In [vdVBS09, vdVBY11] Van de Ven showed that max-weight may fail to provide maximum stability with a dynamic population of flows. The reason for this is that max-weight gets diverted to serving new flows in the network while neglecting a persistent number of flows with relatively small backlog. Similar issues are addressed by Çelik *et al.* in [ÇLM12, ÇM15], where they show that max-weight fails to stabilise a system when there are reconfiguration delays associated to selecting new actions/configurations. Their approach to solve the problem is to use a variable frame size max-weight, where actions are allocated in frames in order to minimise the penalties associated with configuration changes. A similar issue appears in the work in [MSY12] by Maguluri in the context of cloud computing. However, in their case they need to force "inefficiencies" or refresh times in order to choose a max-weight schedule that can change the configuration in a server. The latter work was extended in

[MS14] to allow choosing a max-weight schedule without forcing refresh times. Another case where standard max-weight approaches fail to stabilise a system is with heavy-tailed traffic [NWZ13]. Markakis showed in [MMT14] that max-weight is unstable in the presence of a single heavy-tailed flow, and developed a max-weight extension that stabilises a system in the presence of heavy-tailed traffic for a single hop network. The latter work is extended in [MMT16] to multi-hop networks. Meyn presented in [Mey09] a class of h-MaxWeight policies that have the same desired properties than the classic max-weight, but also support heavy-tailed traffic.

There are other works apart from max-weight in the literature of packet scheduling, *e.g.* Maximal Scheduling [CKLS08] or distributed greedy approaches [WSP07]. However, these only achieve a fraction of the network capacity region. A notable exception is the Longest-Queue-First (LQF) algorithm [AD06], that under a local pooling condition (a particular structure of the graph/network) is able to achieve the full capacity region. Moreover, and unlike max-weight, LQF is not a centralised algorithm and can be used to make distributed and asynchronous scheduling decisions. The LQF is extended to consider packets with variable packet size in [MHS14], and to wireless networks in [LBX11]. Other important algorithms include the Q-CSMA [NTS12], which schedules packets in Carrier Sense Multiple Access (CSMA) networks while taking collisions explicitly into account; and the work in [BC12] by Bonald, which considers asynchronous scheduling schemes for input-queue packet switches with variable packet sizes.

With regard to the existence of a connection between the discrete-valued queue occupancy in a queueing network and continuous-valued Lagrange multipliers, this has been noted by several authors, see for example [LSS06], but we are aware of few rigorous results. A notable exception is [HN11], which establishes that in a deterministic setting a discrete queue update tends on average to drift towards the optimal multiplier value. Also, the *greedy primal-dual* algorithm presented in [Sto05] shows that asymptotically as design parameter $\beta \to 0$ and $t \to \infty$ the scaled queue occupancy converges to the set of dual optima.

Convex Optimisation. Subgradient methods for solving (non-differentiable) convex problems have been studied extensively under various step size rules by Polyak [Pol77], Ermoliev [Erm66] and Shor [Sho12], or more recently by Bertsekas [BNO03] and Nedić [Ned02]. One of the main characteristics of the subgradient methods is that they require to assume very little about the objective functions and constraints, and that they are simple to implement in practice. However, on the other hand, they have slow convergence rate.

Primal averaging in a primal-dual subgradient method was firstly studied by Nemirovski and Yudin in [NY78], and later by Shor [Sho85], Larson [LL97], and Nedić [NO09a, NO09b]. The work in [NO09a] assumes that the dual function can be computed efficiently, and the work in [NO09b] considers a sequence of primal-dual subgradient updates. The inexact computation of the subgradient has been treated in previous work in terms of ϵ_k-subgradients [Ber99, BNO03] or deterministic noise [NB10] in the dual updates. Regarding stochastic (sub)gradients, these are well known in the literature of unconstrained minimisation [Pre95, Mar05], and there is recent work by Ram [RNV09] and Duchi [DHS11] with applications to distributed computing and machine learning.

Within the field of convex optimisation Lagrange multipliers have been regarded in different ways,[1] but the two main views are the following. The first one regards Lagrange multipliers as (static) penalties or auxiliary variables used in iterative methods. This approach is in spirit very similar to von Neumann's minimax game approach in linear programming, which was later generalised by Sion to quasi-concave quasi-convex games [S$^+$58, Kom88]. The second view is that Lagrange multipliers are dual variables arising from a canonical perturbation on the constraints in conjugate duality (see Chapter 11 in [RW98] or SIAM review [Roc74]). The second approach plays an important role in convex optimisation algorithms such as ADMM [BPC$^+$11] or proximal methods [PB14].

Selection of a sequence of actions in a discrete-like manner is also considered in the convex optimisation literature. The *nonlinear Gauss-Seidel* algorithm, also known as *block coordinate descent* [Ber99, BT89] minimises a convex function over a convex set by updating one co-ordinate at a time. The convex function is required to be continuously differentiable and strictly convex and, unlike in the max-weight algorithms discussed above, the action set is convex. The classical Frank-Wolfe algorithm [FW56] also minimises a convex continuously differentiable function over a polytope by selecting from a discrete set of descent directions, although a continuous-valued line search is used to determine the final update. We also note the work on online convex optimisation [Zin03, FKM05], where the task is to choose a sequence of actions so to minimise an unknown sequence of convex functions with low regret.

1.5. Thesis Outline

The structure of this thesis is organised in four major chapters. We start in Chapter 2 by studying how discrete actions fit into convex optimisation from a primal-dual approach and by relating a sequence of convex optimisations with non-convex actions. The presented approach is in spirit very similar to the dynamical system approach used by Kelly in

[1] See, for example, Rockafellar's review in [Roc93] about Lagrange multipliers and optimality conditions.

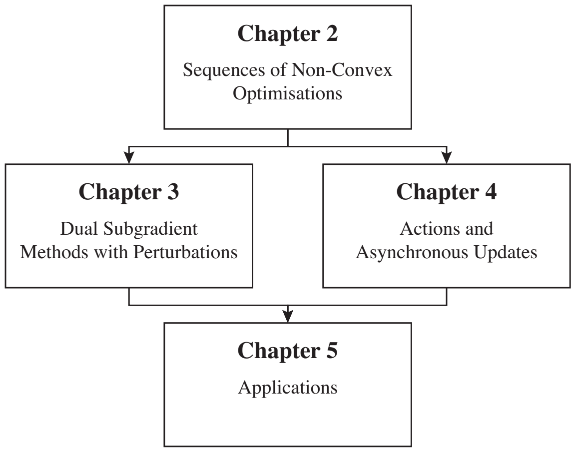


FIGURE 1.5.1. Structure of the thesis.

[KMT98], or Stolyar's greedy primal-dual algorithm [Sto05]. In each iteration we aim to find a discrete direction (a point selected from a non-convex set) that ensures primal descent, and then control the ratio between the primal and dual step sizes in order to have dual ascent. The major novelties in the chapter are the use of non-convex points (actions) as descent directions and to establish a connection between Lagrange multipliers and queue occupancies. The approach has however limitations. Unlike most max-weight approaches (including Tassiulas' [TE92]) it is deterministic and its extension to handle stochasticity is not straightforward. Furthermore, the choice of discrete action used to ensure descent turns out at the end to be unnecessary in terms of optimality conditions. In other words, we realise that it is possible to capture max-weight features (including stochasticity) from a simple dual approach with elementary perturbations: stochastic updates and ϵ_k-subgradients.

Taking these two observations into account, we extend Chapter 2 via two chapters (see Figure 1.5.1). Chapter 3 deals with the convergence of the dual subgradient method with perturbations (δ_k and ϵ_k) in a general form, and Chapter 4 is devoted to studying how the ϵ_k perturbations can be used to capture discrete actions in max-weight. Chapter 4 includes as well how to use perturbations on the Lagrange multipliers (approximate Lagrange multipliers) in order to obtain asynchronous dual updates in an easy manner.

We conclude the thesis in Chapter 5 by putting together the results in Chapters 3 and 4 with some applications. In the first application we provide a summary (or simplified version) of Chapters 3 and 4 while considering a traffic signal control example. The other two applications are presented in a concise manner and show how the convex optimisation framework can be used to easily handle non-convex stochastic network optimisations problems.

CHAPTER 2

Sequences of Non-Convex Optimisations

In brief, we consider a queueing network where the queue occupancy of the j'th queue at time k is denoted by $Q_k(j) \in \mathbf{N}$, $j = 1, 2, \ldots, n$, and we gather these together into vector $Q_k \in \mathbf{N}^n$. Time is slotted and at each time step $k = 1, 2, \ldots$ we select action $y_k \in Y \subset \mathbf{N}^n$, *e.g.* selecting j'th element $x_k(j) = 1$ corresponds to transmitting one packet from queue j and $x_k(j) = 0$ to doing nothing. The connectivity between queues is captured via matrix $A \in \{-1, 0, 1\}^{n \times n}$, whose j'th row has a -1 at the j'th entry, 1 at entries corresponding to queues from which packets are sent to, and 0 entries elsewhere. The queue occupancy then updates according to $Q_{k+1} = [Q_k + Ay_k + b_k]^+$, where the j'th element of vector $b_k \in \mathbf{N}^n$ denotes the number of external packet arrivals to queue j at time k. The objective is to stabilise all of the queues in the system while maximising a concave utility $U : \mathbf{R}^n \to \mathbf{R}$ of the running average of the discrete actions y_i, $i = 1, \ldots, k$.

In this chapter, we investigate the connections between max-weight approaches and dual subgradient methods in convex optimisation, and show that strong connections do indeed exist. We take as starting point the *greedy primal-dual* variant of max-weight scheduling used by Stolyar [Sto05], which selects action $y_k \in \arg\max_{y \in Y} \partial U(x_k)^T y - \beta Q_k^T A y$ with $x_{k+1} = (1 - \beta) x_k + \beta y_k$ and $0 < \beta < 1$ is a design parameter.

2.1. Preliminaries

Recall the following convexity properties.

LEMMA 2.1 (Lipschitz Continuity). *Let $\psi : M \to \mathbf{R}$ be a convex function and let X be a closed and bounded set contained in the relative interior of the domain $M \subseteq \mathbf{R}^n$. Then ψ is Lipschitz continuous on X* i.e. *there exists a constant ν_ψ such that $|\psi(x_1) - \psi(x_2)| \le \nu_\psi \|x_1 - x_2\|_2 \ \forall x_1, x_2 \in X$.*

PROOF. See, for example, [RV74]. ■

LEMMA 2.2 (Bounded Distance). *Let $Y := \{y_1, \ldots, y_{|Y|}\}$ be a finite set of points from $\mathbf{R}^n$. Then there exists constant $y_\circ$ such that $\|x_1 - x_2\|_2 \le y_\circ$ for any two points $x_1, x_2 \in X := \mathrm{conv}(Y)$, where $\mathrm{conv}(Y)$ denotes the convex hull of Y.*

PROOF. Since $x_1, x_2 \in X$ these can be written as the convex combination of points in Y, *i.e.*

$$x_1 = \sum_{j=1}^{|Y|} \theta(j) y_j, \quad x_2 = \sum_{j=1}^{|Y|} \tilde{\theta}(j) y_j$$

with $\theta, \tilde{\theta} \succeq 0$, and $\|\theta\|_1 = \|\tilde{\theta}\|_1 = 1$. Hence, $\|x_1 - x_2\|_2 = \| \sum_{j=1}^{|Y|}(\theta(j) - \tilde{\theta}(j)) y_j\|_2 \leq \sum_{j=1}^{|Y|} \|\theta(j) - \tilde{\theta}(j)\|_2 \|y_j\|_2 \leq 2 \max_{y \in Y} \|y\|_2 := y_\circ$. ■

We also introduce the following definition:

DEFINITION 2.1 (Bounded Curvature). *Let $\psi : M \to \mathbf{R}$ be a convex function defined on domain $M \subseteq \mathbf{R}^n$. We say ψ has bounded curvature on set $X \subset M$ if for any points $x, x + \delta \in X$*

$$\psi(x + \delta) - \psi(x) \leq \partial\psi(x)^T \delta + \rho_\psi \|\delta\|_2^2 \tag{2.1}$$

where $\rho_\psi \geq 0$ is a constant that does not depend on x or δ.

Bounded curvature will prove important in our analysis. The following lemma shows that a necessary and sufficient condition for bounded curvature is that the subgradients of ψ are Lipschitz continuous on set X.

LEMMA 2.3 (Bounded Curvature). *Let $\psi : M \to \mathbf{R}$, $M \subseteq \mathbf{R}^n$ be a convex function. Then ψ has bounded curvature on X if and only if for all $x, x+\delta \in X$ there exists a member $\partial\psi(x)$ (respectively, $\partial\psi(x+\delta)$) of the set of subdifferentials at point x (respectively, $x+\delta$) such that*

$$(\partial\psi(x + \delta) - \partial\psi(x))^T \delta \leq \rho_\psi \|\delta\|_2^2$$

where ρ_ψ does not depend on x or δ.

PROOF. $\Rightarrow$ Suppose ψ has bounded curvature on X. From (2.1) it follows that

$$\psi(x + \delta) - \psi(x) \leq \partial\psi(x)^T \delta + \rho_\psi \|\delta\|_2^2$$

and

$$\psi(x) - \psi(x + \delta) \leq -\partial\psi(x + \delta)^T \delta + \rho_\psi \|\delta\|_2^2.$$

Adding the left-hand and right-hand sides of these inequalities yields $0 \leq (\partial\psi(x) - \partial\psi(x+\delta))^T \delta + 2\rho_\psi \|\delta\|_2^2$ *i.e.*

$$(\partial\psi(x + \delta) - \partial\psi(x))^T \delta \leq \rho_\psi \|\delta\|_2^2.$$

$\Leftarrow$ Suppose $(\partial\psi(x+\delta)-\partial\psi(x))^T\delta \le \rho_\psi\|\delta\|_2$ for all $x, x+\delta \in M$. It follows that

$$\partial\psi(x+\delta)^T\delta \le \partial\psi(x)^T\delta + \rho_\psi\|\delta\|_2^2.$$

By the definition of the subgradient we have that $\psi(x+\delta)-\psi(x) \le \partial\psi(x+\delta)^T\delta$, and so we obtain that $\psi(x+\delta)-\psi(x) \le \partial\psi(x)^T\delta + \rho_\psi\|\delta\|_2^2$. ■

2.2. Non-Convex Descent

We begin by considering the minimisation of convex function $F : \mathbf{R}^n \to \mathbf{R}$ on convex set $X := \text{conv}(Y)$, the convex hull of set $Y := \{y_1, \dots, y_{|Y|}\}$ consisting of a finite collection of points from $\mathbf{R}^n$ (so X is a polytope). Our interest is in selecting a sequence of points $\{y_k\}$, $k = 1, 2, \dots$ from set Y such that the running average $x_{k+1} = (1-\beta)x_k + \beta y_k$ minimises F for k sufficiently large and β sufficiently small. Note that set Y is non-convex since it consists of a finite number of points, and by analogy with max-weight terminology we will refer to it as the *action set*.

Since X is the convex hull of action set Y, any point $s^* \in X$ minimising F can be written as convex combinations of points in Y *i.e.* $s^* = \sum_{j=1}^{|Y|}\theta^*(j)y_j$, $\theta^*(j) \in [0,1]$, $\|\theta\|_1 = 1$. Hence, we can always construct sequence $\{y_k\}$ by selecting points from set Y in proportion to the $\theta^*(j)$, $j = 1, \dots, |Y|$—that is, by *a posteriori* time-sharing (*a posteriori* in the sense that we need to find minimum s^* before we can construct sequence $\{y_k\}$). Of more interest, however, it turns out that when function F has bounded curvature then sequences $\{y_k\}$ can be found without requiring knowledge of s^*.

2.2.1. Non-Convex Direct Descent.

The following two fundamental results are the key to establishing Theorem 2.1.

LEMMA 2.4. *Let $Y := \{y_1, \dots, y_{|Y|}\}$ be a finite set of points from $\mathbf{R}^n$ and $X :=$ conv(Y). Then, for any point $x \in X$ and vector $z \in \mathbf{R}^n$ there exists a point $y \in Y$ such that $z^T(y-x) \le 0$.*

PROOF. Since $x \in X := \text{conv}(Y)$, $x = \sum_{j=1}^{|Y|}\theta(j)y_j$ with $\sum_{j=1}^{|Y|}\theta(j) = 1$ and $\theta(j) \in [0,1]$. Hence,

$$z^T(y-x) = \sum_{j=1}^{|Y|}\theta(j)z^T(y-y_j).$$

Select $y \in \arg\min_{w\in Y} z^T w$. Then $z^T y \le z^T y_j$ for all $y_j \in Y$ and so $z^T(y-x) \le 0$. ■

LEMMA 2.5 (Non-Convex Descent). *Let $F(x)$ be a convex function and suppose points z, $x \in X = \text{conv}(Y)$ exist such that $F(z) \le F(x) - \epsilon$, $\epsilon > 0$. Suppose F has bounded* *curvature on X with curvature constant ρ_F. Then, there exists at least one $y \in Y$ such that*

$$F((1-\beta)x + \beta y) \leq F(x) - \gamma\beta\epsilon$$

with $\gamma \in (0,1)$ provided $0 < \beta \leq (1-\gamma)\min\{\epsilon/(\rho_F y_\circ^2), 1\}$.

Proof. By convexity,

$$F(x) + \partial F(x)^T(z-x) \leq F(z) \leq F(x) - \epsilon.$$

Hence, $\partial F(x)^T(z-x) \leq -\epsilon$. Now observe that for $y \in Y$ we have $(1-\beta)x + \beta y \in X$ and by the bounded curvature of F on X

$$\begin{aligned}
&F((1-\beta)x + \beta y) \\
&\leq F(x) + \beta\partial F(x)^T(y-x) + \rho_F\beta^2\|y-x\|_2^2 \\
&= F(x) + \beta\partial F(x)^T(z-x) + \beta\partial F(x)^T(y-z) + \rho_F\beta^2\|y-x\|_2^2 \\
&\leq F(x) - \beta\epsilon + \beta\partial F(x)^T(y-z) + \rho_F\beta^2\|y-x\|_2^2
\end{aligned}$$

By Lemma 2.4 we can select $y \in Y$ such that $\partial F(x)^T(y-z) \leq 0$. With this choice of y it follows that

$$\begin{aligned}
F((1-\beta)x + \beta y) &\leq F(x) - \beta\epsilon + \rho_F\beta^2\|y-x\|_2^2 \\
&\leq F(x) - \beta\epsilon + \rho_F\beta^2 y_\circ^2 \qquad (2.2)
\end{aligned}$$

where (2.2) follows from Lemma 2.2, and the result now follows. ■

The following theorem formalises the above commentary, also generalising it to sequences of convex functions $\{F_k\}$ rather than just a single function as this will prove useful later.

Theorem 2.1 (Greedy Non-Convex Convergence). *Let $\{F_k\}$ be a sequence of convex functions with uniformly bounded curvature ρ_F on set $X := \mathrm{conv}(Y)$, action set Y a finite set of points from $\mathbf{R}^n$. Let $\{x_k\}$ be a sequence of vectors satisfying $x_{k+1} = (1-\beta)x_k + \beta y_k$ with $x_1 \in X$ and*

$$y_k \in \arg\min_{y \in Y} F_k((1-\beta)x_k + \beta y), \qquad k = 1, 2, \ldots \qquad (2.3)$$

Suppose parameter β is sufficiently small that

$$0 < \beta \leq (1-\gamma)\min\{\epsilon/(\rho_F y_\circ^2), 1\} \qquad (2.4)$$

with $\epsilon > 0$, $\gamma \in (0,1)$, $y_\circ := 2\max_{y\in Y}\|y\|_2$ *and that functions* F_k *change sufficiently slowly that*

$$|F_{k+1}(x) - F_k(x)| \leq \gamma_1\gamma\beta\epsilon,\ \forall x \in X$$

with $\gamma_1 \in (0, 1/2)$*. Then, there exists a* $\bar{k} \in \mathbf{N}$ *such that for all* $k \geq \bar{k}$ *we have that*

$$0 \leq F_k(x_{k+1}) - F_k(s_k^*) \leq 2\epsilon$$

where $s_k^* \in \arg\min_{x\in X} F_k(x)$*.*

PROOF. Since F_k has bounded curvature for any k it is continuous, and as X is closed and bounded we have by the Weierstrass theorem (*e.g.* see Proposition 2.1.1 in [BNO03]) that $\min_{x\in X} F_k(x)$ is finite. We now proceed considering two cases:

Case (i): $F_k(x_k) - F_k(s_k^*) \geq \epsilon$. By Lemma 2.5 there exists $y_k \in Y$ such that

$$F_k((1-\beta)x_k + \beta y_k) - F_k(x_k) = F_k(x_{k+1}) - F_k(x_k) \leq -\gamma\beta\epsilon.$$

Further, since $F_{k+1}(x_{k+1}) - F_k(x_{k+1}) \leq \gamma_1\gamma\beta\epsilon$ and $F_k(x_k) - F_{k+1}(x_k) \leq \gamma_1\gamma\beta\epsilon$ it follows

$$F_{k+1}(x_{k+1}) - F_{k+1}(x_k) \leq 2\gamma_1\gamma\beta\epsilon - \gamma\beta\epsilon < 0. \tag{2.5}$$

That is, F_k and F_{k+1} decrease monotonically when $F_k(x_k) - F_k(s_k^*) \geq \epsilon$.

Case (ii): $F_k(x_k) - F_k(s_k^*) < \epsilon$. It follows that $F_k(x_k) < F_k(s_k^*) + \epsilon$. Since F_k is convex and has bounded curvature,

$$\begin{aligned} F_k(x_{k+1}) &\leq F_k(x_k) + \beta\partial F_k(x_k)^T(y_k - x_k) + \beta^2\rho_F y_\circ^2 \\ &\leq F_k(s_k^*) + \epsilon + \beta\partial F_k(x_k)^T(y_k - x_k) + \beta^2\rho_F y_\circ^2. \end{aligned}$$

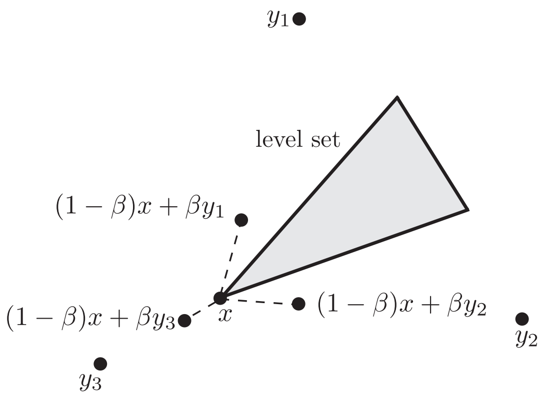
The final term holds uniformly for all $y_k \in Y$ and since we select y_k to minimise $F_k(x_{k+1})$ by Lemma 2.4 we therefore have $F_k(x_{k+1}) \leq F_k(s_k^*) + \epsilon + \beta^2\rho_F y_\circ^2$. Using the stated choice of β and the fact that $F_{k+1}(x_{k+1}) - \gamma_1\gamma\beta\epsilon \leq F_k(x_{k+1})$ yields

$$F_{k+1}(x_{k+1}) - F_k(s_k^*) \leq \gamma_1\gamma\beta\epsilon + \epsilon + \beta(1-\gamma)\epsilon. \tag{2.6}$$

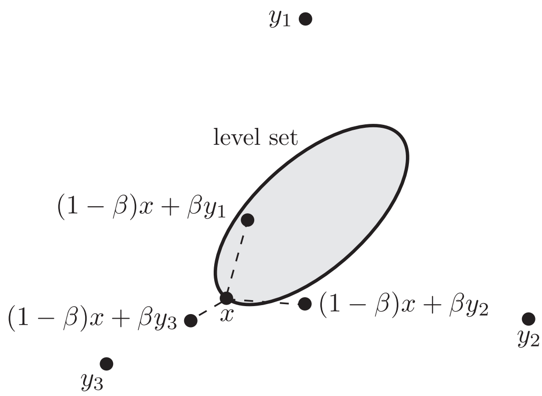
Finally, since $F_k(s_k^*) \leq F_k(s_{k+1}^*) \leq F_{k+1}(s_{k+1}^*) + \gamma_1\gamma\beta\epsilon$ we obtain

$$\begin{aligned} F_{k+1}(x_{k+1}) - F_{k+1}(s_{k+1}^*) &\leq 2\gamma_1\gamma\beta\epsilon + \epsilon + \beta(1-\gamma)\epsilon \\ &\leq 2\epsilon. \end{aligned}$$

We therefore have that $F_{k+1}(x_k)$ is strictly decreasing when $F_k(x_k) - F_k(s_k^*) \geq \epsilon$ and otherwise uniformly upper bounded by 2ϵ. It follows that for all k sufficiently large $F_k(x_{k+1}) - F_k(s_k^*) \leq 2\epsilon$ as claimed. ■



(A) Example where F does not possess bounded curvature. None of $(1-\beta)x+\beta y$, $y \in Y$ are descending.



(B) Example where F has bounded curvature. For β sufficiently small, for at least one $(1-\beta)x+\beta y$, $y \in Y$ descent is possible.

FIGURE 2.2.1. Illustrating how bounded curvature allows monotonic descent. Set Y consists of the marked points y_1, y_2, y_3. Level set $\{F(z) \leq F(x) : z \in X\}$ is indicated by the shaded areas. The possible choices of $(1-\beta)x+\beta y$, $y \in Y$ are indicated.

Observe that in Theorem 2.1 we select y_k by solving non-convex optimisation (2.3) at each time step. This optimisation is one step ahead, or greedy, in nature and does not look ahead to future values of the sequence or require knowledge of optima s_k^*. Of course, such an approach is mainly of interest when non-convex optimisation (2.3) can be efficiently solved, *e.g.* when action set Y is small or the optimisation separable.

Observe also that Theorem 2.1 relies upon the bounded curvature of the sequence of functions F_k. A smoothness assumption of this sort seems essential, since when it does not hold it is easy to construct examples where Theorem 2.1 does not hold. Such an example is illustrated schematically in Figure 2.2.1a. The shaded region in Figure 2.2.1a indicates the level set $\{F(z) \leq F(x) : z \in X\}$. The level set is convex, but the boundary is non-smooth and contains "kinks". We can select points from the set $\{(1-\beta)x+\beta y : y \in Y = \{y_1, y_2, y_3\}\}$. This set of points is indicated in Figure 2.2.1a and it can be seen that every point lies outside the level set. Hence, we must have $F((1-\beta)x+\beta y) > F(x)$, and upon iterating we will end up with a diverging sequence. Note that in this example changing the step size β does not resolve the issue. Bounded curvature ensures that the boundary of the level sets is smooth, and this ensures that for sufficiently small β there exists a convex combination of x with a point $y \in Y$ such that $F((1-\beta)x+\beta y) < F(x)$ and so the solution to optimisation (2.3) improves our objective, see Figure 2.2.1b.

Theorem 2.1 is stated in a fairly general manner since this will be needed for our later analysis. An immediate corollary to Theorem 2.1 is the following convergence result for unconstrained optimisation.

COROLLARY 2.1 (Unconstrained Optimisation). *Consider the following sequence of non-convex optimisations* $\{\mathcal{P}_k\}$*:*

$$y_k \in \arg\min_{y \in Y} f\left((1-\beta)x_k + \beta y\right)$$

$$x_{k+1} = (1-\beta)x_k + \beta y_k$$

with $x_1 \in X := \mathrm{conv}(Y)$*, action set* $Y \subset \mathbf{R}^n$ *finite. Then* $0 \leq f(x_k) - f^\star \leq 2\epsilon$ *for all* k *sufficiently large, where* $f^\star = \min_{x \in X} f(x)$*, provided* f *has bounded curvature with curvature constant* ρ_F *and* $0 < \beta \leq (1-\gamma)\min\{\epsilon/(\rho_F y_\circ^2), 1\}$ *with* $\gamma \in (0,1)$*,* $\epsilon > 0$*,* $y_\circ := 2\max_{y \in Y} \|y\|_2$*.*

Figure 2.2.2 illustrates Corollary 2.1 schematically in $\mathbf{R}^2$. The sequence of non-convex optimisations descends in two iterations $f(x_1) > f(x_2) > f(x_3)$ (using points y_3 and y_4 respectively) and $f(x_k) - f^\star \leq 2\epsilon$ for $k > 3$ (not shown in Figure 2.2.2).

Note that the curvature constant ρ_F of function f need not be known, an upper bound being sufficient to select β. Next we present two brief examples that are affected differently by constant ρ_F.

EXAMPLE 2.1 (Linear Objective). *Suppose* $f(x) := A^T x$ *with* $A \in \mathbf{R}^n$*. The objective function is linear and so has curvature constant* $\rho_F = 0$*. It can be seen from (2.4) that we can choose* β *independently of parameter* ϵ*. Hence, for any* $\beta \in (0,1)$ *we have that* $f(x_{k+1}) < f(x_k)$ *for all* k*, and since we could always "choose" a smaller* ϵ *(because* β *does not depend on it) we must have that* $f(x_k) \to f^\star$*.*

EXAMPLE 2.2 (Quadratic Objective). *Suppose* $f(x) := \frac{1}{2}x^T A x$ *where* $A \in \mathbf{R}^{n \times n}$ *is symmetric and positive definite. Then* $\rho_F = \lambda_{\max}(A) > 0$ *and in contrast to Example 2.1 the bound (2.4) on parameter* β *now depends on* ϵ*. Further, the convergence is into the ball* $f(x_k) - f^\star \leq 2\epsilon$ *for* $k \geq \bar{k}$ *and finite* $\bar{k}$*.*

2.2.2. Non-Convex Frank-Wolfe-like Descent. It is important to note that other convergent non-convex updates are also possible. For example:

THEOREM 2.2 (Greedy Non-Convex FW Convergence). *Consider the setup in Theorem 2.1, but with modified update*

$$y_k \in \arg\min_{y \in Y} \partial F_k(x_k)^T y, \qquad k = 1, 2, \ldots \tag{2.7}$$

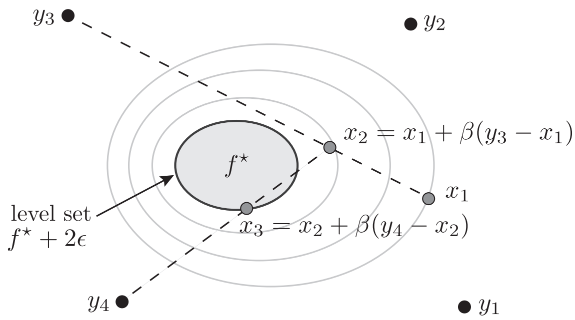


FIGURE 2.2.2. Illustrating unconstrained convergence in $\mathbf{R}^2$. The sequence of non-convex optimisations converges with $k = 2$. The function average decreases monotonically and then remains in level set $f(x_k) \le f^\star + 2\epsilon$ for $k \ge 3$.

Then, there exists a $\bar{k} \in \mathbf{N}$ such that for all $k \ge \bar{k}$ we have that

$$0 \le F_k(x_{k+1}) - F_k(s_k^*) \le 2\epsilon$$

where $s_k^* \in \arg\min_{x\in X} F_k(x)$.

PROOF. Firstly, we make the following observations,

$$\begin{aligned}\arg\min_{x\in X} F_k(x_k) + \partial F_k(x_k)^T(x - x_k) &\overset{(a)}{=} \arg\min_{x\in X} \partial F_k(x_k)^T x \\ &\overset{(b)}{\supseteq} \arg\min_{y\in Y} \partial F_k(x_k)^T y\end{aligned}$$

where equality (a) follows by dropping terms not involving x and (b) from the observation that we have a linear program (the objective is linear and set X is a polytope, so defined by linear constraints) and so the optimum lies at an extreme point of set X *i.e.* in set Y. We also have that

$$\begin{aligned}F_k(x_k) + \partial F_k(x_k)^T(y_k - x_k) &\overset{(a)}{\le} F_k(x_k) + \partial F_k(x_k)^T(s_k^* - x_k) \\ &\overset{(b)}{\le} F_k(s_k^*) \le F_k(x_k)\end{aligned}$$

where $s_k^* \in \arg\min_{x\in X} F_k(x)$, inequality (a) follows from the minimality of y_k in X noted above and (b) from the convexity of F_k. It follows that $\partial F_k(x_k)^T(y_k - x_k) \le -(F_k(x_k) - F_k(s_k^*)) \le 0$. We have two cases to consider. Case (i): $F_k(x_k) - F_k(s_k^*) \ge \epsilon$. By the bounded curvature of F_k,

$$\begin{aligned}F_k(x_{k+1}) &\le F_k(x_k) + \beta\partial F_k(x_k)^T(y_k - x_k) + \rho_F \beta^2 y_\circ^2 \\ &\le F_k(x_k) - \beta\epsilon + \rho_F\beta^2 y_\circ^2 \le F_k(x_k) - \gamma\beta\epsilon.\end{aligned}$$

Hence,

$$F_{k+1}(x_{k+1}) \leq F_k(x_{k+1}) + |F_{k+1}(x_{k+1}) - F_k(x_{k+1})|$$
$$\leq F_k(x_k) - \gamma\beta\epsilon + \gamma_1\gamma\beta\epsilon,$$

and since $F_k(x_k) \leq F_{k+1}(x_k) + \gamma_1\gamma\beta\epsilon$ we have that $F_{k+1}(x_{k+1}) - F_k(x_{k+1}) < 0$. Case (ii): $F_k(x_k) - F_k(s_k^*) < \epsilon$. Then

$$F_k(x_{k+1}) \leq F_k(x_k) + \beta\partial F_k(x_k)^T(y_k - x_k) + \rho_F\beta^2 y_\circ^2$$
$$\leq F_k(s_k^*) + \epsilon + \beta\epsilon,$$

and similar to the proof of Theorem 2.1 we obtain that $F_{k+1}(x_{k+1}) - F_{k+1}(s_{k+1}^*) \leq 2\epsilon$. We therefore have that $F_k(x_k)$ is strictly decreasing when $F_k(x_k) - F_k(s_k^*) \geq \epsilon$ and otherwise uniformly upper bounded by 2ϵ. Thus for k sufficiently large $F_k(x_{k+1}) - F_k(s_k^*) \leq 2\epsilon$. ■

The intuition behind the update in Theorem 2.2 is that at each step we locally approximate $F_k(x_{k+1})$ by linear function $F_k(x_k) + \partial F_k(x_k)^T(x_{k+1} - x_k)$ and then minimise this linear function. Since F_k is convex, this linear function is in fact the supporting hyperplane to F_k at point x_k, and so can be expected to allow us to find a descent direction. Similar intuition also underlies classical Frank-Wolfe algorithms for convex optimisation [FW56] on a polytope, and Theorem 2.2 extends this class of algorithms to make use of non-convex update (2.7) and a fixed step size (rather than the classical approach of selecting the step size by line search).

Note that when the function is linear $F_k(x) = c_k^T x$, $c_k \in \mathbf{R}^n$, then

$$\arg\min_{y \in Y} F_k((1-\beta)x + \beta y) = \arg\min_{y \in Y} c_k^T y \tag{2.8}$$

and

$$\arg\min_{y \in Y} \partial F_k(x_k)^T y = \arg\min_{y \in Y} c_k^T y. \tag{2.9}$$

That is, updates (2.3) and (2.7) are identical. Note also that

$$\arg\min_{y \in Y} \partial F_k(x_k)^T y \subseteq \arg\min_{x \in X} \partial F_k(x_k)^T x. \tag{2.10}$$

This is because the RHS of (2.10) is a linear programme (the objective is linear and set X is a polytope, so defined by linear constraints) and so the optimum set is either (i) an extreme point of X and so a member of set Y, or (ii) a face of polytope X with the extreme points of the face belonging to set Y. Hence, while update (2.7) is non-convex it can nevertheless be solved in polynomial time.

2.3. Constrained Convex Optimisation

We now extend consideration to the constrained convex optimisation $\mathcal{P}$:

$$\begin{aligned} &\underset{x\in X}{\text{minimise}} && f(x) \\ &\text{subject to} && g(x) \preceq 0 \end{aligned} \tag{2.11}$$

where $g(x) := [g_1, \dots, g_m]^T$ and $f, g_j : \mathbf{R}^n \to \mathbf{R}$, $j = 1, \dots, m$ are convex functions with bounded curvature with, respectively, curvature constants ρ_F and ρ_{g_j}. As before, action set Y consists of a finite set of points in $\mathbf{R}^n$ and $X := \text{conv}(Y)$. Let $X_0 := \{x \in X \mid g(x) \preceq 0\}$ denote the set of feasible points, which we will assume has non-empty relative interior (*i.e.* a Slater point exists). Let $X^\star := \arg\min_{x\in X_0} f(x) \subseteq X_0$ be the set of optima and $f^\star := f(x^\star)$, $x^\star \in X^\star$.

In the next sections we introduce a generalised dual subgradient approach for finding approximate solutions to optimisation $\mathcal{P}$ which, as we will see, includes the classical convex dual subgradient method as a special case.

2.3.1. Lagrangian Penalty. As in classical convex optimisation we define Lagrangian $L(x,\lambda) := f(x) + \lambda^T g(x)$ where $\lambda = [\lambda(1), \dots, \lambda(m)]^T$ with $\lambda(j) \geq 0$, $j = 1, \dots, m$. Since set X_0 has non-empty relative interior, the Slater condition is satisfied and strong duality holds. That is, there is zero duality gap and so the solution of the dual problem $\mathcal{D}$:

$$\underset{\lambda\succeq 0}{\text{maximise}} \quad h(\lambda) := \min_{x\in X} L(x,\lambda)$$

and primal problem $\mathcal{P}$ coincide. Therefore, we have that

$$\min_{x\in X}\max_{\lambda\succeq 0} L(x,\lambda) = \max_{\lambda\succeq 0}\min_{x\in X} L(x,\lambda) = h(\lambda^\star) = f^\star$$

where $\lambda^\star := \arg\max_{\lambda\succeq 0} h(\lambda)$.

2.3.1.1. *Lagrangian Bounded Curvature.* As already noted, bounded curvature plays a key role in ensuring convergence to an optimum when selecting from a discrete set of actions. For any two points $x, x+\delta \in X$ we have that

$$L(x+\delta,\lambda) \leq L(x,\lambda) + \partial_x L(x,\lambda)^T\delta + \rho_L\|\delta\|_2^2,$$

where $\rho_L = \rho_F + \lambda^T\rho_g$ with $\rho_g := [\rho_{g_1}, \dots, \rho_{g_m}]^T$. It can be seen that the curvature constant ρ_L of the Lagrangian depends on the multiplier λ. Since set $\lambda \succeq 0$ is unbounded, it follows that the Lagrangian does not have bounded curvature on this set unless $\rho_g = 0$ (corresponding to the special case where the constraints are linear). Fortunately, by constraining $\lambda(j) \leq \lambda^\circ$, $j = 1, \dots, m$ for some $\lambda^\circ \geq 0$ resolves the issue, *i.e.* now $L(\cdot,\lambda)$

has uniform bounded curvature with constant

$$\bar{\rho}_L = \rho_F + \lambda^\circ \|\rho_g\|_1.$$

For bounded curvature we only require constant λ° to be finite, but as we will see later in Lemmas 2.7 and 2.9 in general it should be chosen with some care.

2.3.2. Non-Convex Dual Subgradient Update. In this section we present a primal-dual-like approach in which we use discrete actions to obtain approximate solutions to problem $\mathcal{P}$. In particular, we construct a sequence $\{x_k\}$ of points in X such that $f(\frac{1}{k}\sum_{i=1}^{k} x_{i+1})$ is arbitrarily close to $f^\star$ for k sufficiently large.

We start by introducing two lemmas, which will play a prominent role in later proofs.

LEMMA 2.6 (Minimising Sequence of Lagrangians). *Let $\{\lambda_k\}$ be a sequence of vectors in $\mathbf{R}^m_+$ such that $\lambda_k \preceq \lambda^\circ \mathbf{1}$, $\lambda^\circ > 0$ and $\|\lambda_{k+1} - \lambda_k\|_2 \leq \gamma_1 \gamma \beta \epsilon / (m \sigma_c)$ with $\gamma \in (0,1)$, $\gamma_1 \in (0, 1/2)$, β, $\epsilon > 0$, $\sigma_c := \max_{x \in X} \|g(x)\|_\infty$. Consider optimisation problem $\mathcal{P}$ and updates*

$$y_k \in \arg\min_{y \in Y} L((1-\beta)x_k + \beta y, \lambda_k), \tag{2.12}$$

$$x_{k+1} = (1-\beta)x_k + \beta y_k. \tag{2.13}$$

Then, for k sufficiently large ($k \geq \bar{k}$) we have that

$$L(x_{k+1}, \lambda_k) - h(\lambda_k) \leq L(x_{k+1}, \lambda_k) - f^\star \leq 2\epsilon$$

provided β is sufficiently small, i.e. $0 < \beta \leq (1-\gamma) \min\{\epsilon / (\bar{\rho}_L y_\circ^2), 1\}$ *where* $y_\circ := 2 \max_{y \in Y} \|y\|_2$, $\bar{\rho}_L = \rho_F + \lambda^\circ \|\rho_g\|_1$.

PROOF. Observe that since

$$\begin{aligned} |L(x, \lambda_{k+1}) - L(x, \lambda_k)| &= |(\lambda_{k+1} - \lambda_k)^T g(x)| \\ &\leq \|\lambda_{k+1} - \lambda_k\|_2 \|g(x)\|_2 \\ &\leq \|\lambda_{k+1} - \lambda_k\|_2 m \sigma_c \\ &\leq \gamma_1 \gamma \beta \epsilon \end{aligned}$$

and $L(\cdot, \lambda_k)$ has uniformly bounded curvature by Theorem 2.1 we have that for k sufficiently large ($k \geq \bar{k}$) then

$$L(x_{k+1}, \lambda_k) - h(\lambda_k) \leq 2\epsilon$$

where $h(\lambda) := \min_{x \in X} L(x, \lambda)$. Further, since $h(\lambda) \leq h(\lambda^\star) \leq f^\star$ for all $\lambda \succeq 0$ it follows that

$$L(x_{k+1}, \lambda_k) - f^\star \leq 2\epsilon$$

for $k \geq \bar{k}$. ■

Lemma 2.7 (Lagrangian of Averages). *Consider optimisation problem $\mathcal{P}$ and update*

$$\lambda_{k+1} = [\lambda_k + \alpha g(x_{k+1})]^{[0,\lambda^\circ]}$$

where $\alpha > 0$ and $\{x_k\}$ is a sequence of points from X such that $L(x_{k+1}, \lambda_k) - h(\lambda_k) \leq 2\epsilon$ for all $k = 1, 2, \ldots$. Let $\lambda_1(j) \in [0, \lambda^\circ]$ where $\lambda^\circ \geq \lambda^\star(j)$, $j = 1, \ldots, m$, where $\lambda^\star \in \Lambda^\star$ (the set of dual optima). Then,

$$|L(\bar{x}_k, \bar{\lambda}_k) - f^\star| \leq 2\epsilon + \frac{\alpha}{2} m \sigma_c^2 + \frac{m \lambda^{\circ 2}}{\alpha k} \tag{2.14}$$

where $\bar{x}_k := \frac{1}{k} \sum_{i=1}^k x_{i+1}$, $\bar{\lambda}_k := \frac{1}{k} \sum_{i=1}^k \lambda_i$ and $\sigma_c := \max_{x \in X} \|g(x)\|_\infty$.

Proof. Let $\theta \in \mathbf{R}_+^m$ such that $\theta(j) \leq \lambda^\circ$ for all $j = 1, \ldots, m$ and see that

$$\begin{aligned}
\|\lambda_{k+1} - \theta\|_2^2 &= \|[\lambda_k + \alpha g(x_{k+1})]^{[0,\lambda^\circ]} - \theta\|_2^2 \\
&\leq \|[\lambda_k + \alpha g(x_{k+1})]^+ - \theta\|_2^2 && (2.15) \\
&\leq \|\lambda_k + \alpha g(x_{k+1}) - \theta\|_2^2 \\
&= \|\lambda_k - \theta\|_2^2 + 2\alpha(\lambda_k - \theta)^T g(x_{k+1}) + \alpha^2 \|g(x_{k+1})\|_2^2 \\
&\leq \|\lambda_k - \theta\|_2^2 + 2\alpha(\lambda_k - \theta)^T g(x_{k+1}) + \alpha^2 m \sigma_c^2, && (2.16)
\end{aligned}$$

where (2.15) follows since $\lambda^\circ \geq \theta(j)$ and (2.16) from the fact that $\|g(x)\|_2^2 \leq m\sigma_c^2$ for all $x \in X$. Applying the latter argument recursively for $i = 1, \ldots, k$ yields $\|\lambda_{k+1} - \theta\|_2^2 \leq \|\lambda_1 - \theta\|_2^2 + 2\alpha \sum_{i=1}^k (\lambda_i - \theta)^T g(x_{i+1}) + \alpha^2 m \sigma_c^2 k$. Rearranging terms, dividing by $2\alpha k$, and using the fact that $\|\lambda_{k+1} - \theta\|_2^2 \geq 0$ and $\|\lambda_1 - \theta\|_2^2 \leq 2m\lambda^{\circ 2}$ we have

$$\begin{aligned}
-\frac{m\lambda^{\circ 2}}{\alpha k} - \frac{\alpha}{2} m \sigma_c^2 &\leq \frac{1}{k} \sum_{i=1}^k (\lambda_i - \theta)^T g(x_{i+1}) && (2.17) \\
&= \frac{1}{k} \sum_{i=1}^k L(x_{i+1}, \lambda_i) - L(x_{i+1}, \theta). && (2.18)
\end{aligned}$$

Next, see that by the definition of sequence $\{x_k\}$ we can write $\frac{1}{k} \sum_{i=1}^k L(x_{i+1}, \lambda_i) \leq \frac{1}{k} \sum_{i=1}^k h(\lambda_i) + 2\epsilon \leq h(\bar{\lambda}_k) + 2\epsilon$ where the last inequality follows by the concavity of

h. That is,

$$-\frac{m\lambda^{\circ 2}}{\alpha k} - \frac{\alpha}{2} m\sigma_c^2 - 2\epsilon \leq h(\bar{\lambda}_k) - \frac{1}{k}\sum_{i=1}^{k} L(x_{i+1}, \theta) \tag{2.19}$$

By fixing θ to $\lambda^\star$ and $\bar{\lambda}_k$ and using the fact that $\frac{1}{k}\sum_{i=1}^{k} L(x_{i+1}, \bar{\lambda}_k) \geq L(\bar{x}_k, \bar{\lambda}_k)$ for all $k = 1, 2, \dots$ and $\frac{1}{k}\sum_{i=1}^{k} L(x_{i+1}, \lambda^\star) \geq f^\star$ we have that

$$-\frac{m\lambda^{\circ 2}}{\alpha k} - \frac{\alpha}{2} m\sigma_c^2 - 2\epsilon \leq h(\bar{\lambda}_k) - f^\star \leq 0 \tag{2.20}$$

and

$$-\frac{m\lambda^{\circ 2}}{\alpha k} - \frac{\alpha}{2} m\sigma_c^2 - 2\epsilon \leq h(\bar{\lambda}_k) - L(\bar{x}_k, \bar{\lambda}_k) \leq 0. \tag{2.21}$$

Multiplying (2.20) by -1 and combining it with (2.21) yields the result. ∎

Note that by selecting α sufficiently small in Lemma 2.7 we can obtain a sequence $\{\lambda_k\}$ that changes sufficiently slowly so to satisfy the conditions of Lemma 2.6. Further, by Lemma 2.6 we can construct a sequence of primal variables that satisfy the conditions of Lemma 2.7 for $k \geq \bar{k}$ and it then follows that (2.14) is satisfied.

Lemma 2.7 requires that $\lambda^\star(j) \leq \lambda^\circ$ for all $j = 1, \dots, m$, so it naturally arises the question as to when $\lambda^\star(j)$ (and so λ°) is bounded. This is clarified in the next lemma, which corresponds to Lemma 1 in [NO09a].

Lemma 2.8 (Bounded Multipliers). *Let $Q_\delta := \{\lambda \succeq 0 : h(\lambda) \geq h(\lambda^\star) - \delta\}$ with $\delta \geq 0$ and let the Slater condition hold,* i.e. *there exists a vector $\hat{x} \in X$ such that $g(\hat{x}) \prec 0$. Then, for every $\lambda \in \Lambda_\delta$ we have that*

$$\|\lambda\|_2 \leq \frac{1}{\upsilon}(f(\hat{x}) - h(\lambda^\star) + \delta) \tag{2.22}$$

where $\upsilon := \min_{j \in \{1,\dots,m\}} -g_j(\hat{x})$.

Proof. First of all recall that since the Slater condition holds we have strong duality, *i.e.* $h(\lambda^\star) = f^\star$, and $f^\star$ is finite by Proposition 2.1.1. in [BNO03]. Now observe that when $\lambda \in \Lambda_\delta$ then

$$h(\lambda^\star) - \delta \leq h(\lambda) = \min_{x \in X} L(x, \lambda) \leq f(\hat{x}) + \lambda^T g(\hat{x}),$$

and rearranging terms we obtain

$$-\lambda^T g(\hat{x}) = -\sum_{j=1}^{m} \lambda(j) g_j(\hat{x}) \leq f(\hat{x}) - h(\lambda^\star) + \delta.$$

Next, since $\lambda \succeq 0$ and $-g_j(\hat{x}) > 0$ for all $j = 1, \dots, m$, let $\upsilon := \min_{j \in \{1,\dots,m\}} -g_j(\hat{x})$ and see that $\upsilon \sum_{j=1}^{m} \lambda(j) \leq f(\hat{x}) - h(\lambda^\star) + \delta$. Finally, dividing by υ and using the fact that $\|\lambda\|_2 \leq \sum_{j=1}^{m} \lambda(j)$ the stated result follows. ■

From Lemma 2.8 we have that it is sufficient for X_0 to have non-empty relative interior in order for Λ_δ to be a bounded subset in $\mathbf{R}^m_+$, and since by definition $\lambda^\star \in \Lambda_\delta$ then $\lambda^\star$ is bounded. The bound obtained in Lemma 2.8 depends on $h(\lambda^\star) = f^\star$, which is usually not known. Nevertheless, we can obtain a looser bound if we use the fact that $-h(\lambda^\star) \leq -h(\lambda)$ for all $\lambda \succeq 0$. That is, for every $\lambda \in \Lambda_\delta$ we have that

$$\|\lambda\|_2 \leq \frac{1}{\upsilon}(f(\bar{x}) - h(\lambda_0) + \delta),$$

where λ_0 is an arbitrary vector in $\mathbf{R}^m_+$.

Hence, when the Slater condition is satisfied the upper and lower bounds in (2.14) are finite and can be made arbitrarily small as $k \to \infty$ by selecting the step size α sufficiently small. Convergence of the average of the Lagrangians does not, of course, guarantee that $f(\bar{x}_k) \to f^\star$ unless we also have complementary slackness, *i.e.* $(\bar{\lambda}_k)^T g(\bar{x}_k) \to 0$. Next we present the following lemma, which is a generalisation of Lemma 3 in [NO09a].

Lemma 2.9 (Complementary Slackness and Feasibility). *Let the Slater condition hold and suppose $\{x_k\}$ is a sequence of points in X and $\{\mu_k\}$ a sequence of points in $\mathbf{R}^m_+$ such that*

(i) $L(x_{k+1}, \mu_k) - h(\mu_k) \leq 2\epsilon$ *for all* k;
(ii) $|\lambda_k(j) - \mu_k(j)| \leq \alpha\sigma_0$, $j = 1, \dots, m$

where $\lambda_{k+1} = [\lambda_k + \alpha g(x_{k+1})]^+$, $\epsilon \geq 0$, $\alpha > 0$, $\sigma_0 \geq 0$. *Suppose also that* $\lambda_1(j) \in [0, \lambda^\circ]$ *with*

$$\lambda^\circ \geq \tfrac{3}{\upsilon}(f(\hat{x}) - h(\lambda^\star) + \delta) + \alpha m \sigma_c$$

where $\delta := \alpha(m\sigma_c^2/2 + m^2\sigma_0\sigma_c) + 2\epsilon$, $\sigma_c := \max_{x \in X} \|g(x)\|_\infty$, $\hat{x}$ *a Slater vector and* $\upsilon := \min_{j \in \{1,\dots,m\}} -g_j(\hat{x})$. *Then,* $\lambda_k(j) \leq \lambda^\circ$ *for all* $k = 1, 2, \dots,$

$$-\frac{m\lambda^{\circ 2}}{2\alpha k} - \frac{\alpha}{2} m\sigma_c^2 \leq (\bar{\lambda}_k)^T g(\bar{x}_k) \leq \frac{m\lambda^{\circ 2}}{\alpha k} \tag{2.23}$$

and

$$g_j(\bar{x}_k) \leq \frac{\lambda^\circ}{\alpha k} \tag{2.24}$$

where $\bar{x}_k := \frac{1}{k}\sum_{i=1}^{k} x_{i+1}$ *and* $\bar{\lambda}_k := \frac{1}{k}\sum_{i=1}^{k} \lambda_i$.

PROOF. We start by showing that updates $[\lambda_k + \alpha g(x_{k+1})]^+$ and $[\lambda_k + \alpha g(x_{k+1})]^{[0,\lambda^\circ]}$ are interchangeable when $L(x_{k+1}, \mu_k)$ is uniformly close to $h(\mu_k)$. First of all see that

$$\begin{aligned}\|\lambda_{k+1} - \lambda^\star\|_2^2 &= \|[\lambda_k + \alpha g(x_{k+1})]^+ - \lambda^\star\|_2^2 \\ &\leq \|\lambda_k + \alpha g(x_{k+1}) - \lambda^\star\|_2^2 \\ &= \|\lambda_k - \lambda^\star\|_2^2 + \alpha^2 \|g(x_{k+1})\|_2^2 + 2\alpha(\lambda_k - \lambda^\star)^T g(x_{k+1}) \\ &\leq \|\lambda_k - \lambda^\star\|_2^2 + \alpha^2 m\sigma_c^2 + 2\alpha(\lambda_k - \lambda^\star)^T g(x_{k+1}).\end{aligned}$$

Now observe that since $\|\lambda_k - \mu_k\|_2 \leq \|\lambda_k - \mu_k\|_1 \leq \alpha m \sigma_0$ we can write

$$\begin{aligned}(\lambda_k - \lambda^\star)^T g(x_{k+1}) &= (\mu_k - \lambda^\star)^T g(x_{k+1}) + (\lambda_k - \mu_k)^T g(x_{k+1}) \\ &\leq (\mu_k - \lambda^\star)^T g(x_{k+1}) + \|\lambda_k - \mu_k\|_2 \|g(x_{k+1})\|_2 \\ &\leq (\mu_k - \lambda^\star)^T g(x_{k+1}) + \alpha m^2 \sigma_0 \sigma_c \\ &= L(x_{k+1}, \mu_k) - L(x_{k+1}, \lambda^\star) + \alpha m^2 \sigma_0 \sigma_c.\end{aligned}$$

Furthermore, since $L(x_{k+1}, \mu_k) \leq h(\mu_k) + 2\epsilon$ and $-L(x_{k+1}, \lambda^\star) \leq -h(\lambda^\star)$ it follows that

$$\begin{aligned}&\|\lambda_{k+1} - \lambda^\star\|_2^2 - \|\lambda_k - \lambda^\star\|_2^2 \\ &\quad \leq \alpha^2(m\sigma_c^2 + 2m^2\sigma_0\sigma_c) + 2\alpha(h(\mu_k) + 2\epsilon - h(\lambda^\star)).\end{aligned} \tag{2.25}$$

Now let $\Lambda_\delta := \{\lambda \succeq 0 : h(\lambda) \geq h(\lambda^\star) - \delta\}$ and consider two cases. Case (i) ($\mu_k \notin Q_\delta$). Then $h(\mu_k) - h(\lambda^\star) < -\delta$ and from (2.25) we have that $\|\lambda_{k+1} - \lambda^\star\|_2^2 < \|\lambda_k - \lambda^\star\|_2^2$, *i.e.*

$$\|\lambda_{k+1} - \lambda^\star\|_2 - \|\lambda_k - \lambda^\star\|_2 < 0$$

and so λ_k converges to a ball around $\lambda^\star$ when $\mu_k \in \Lambda_\delta$. Case (ii) ($\mu_k \in Q_\delta$). Observe that

$$\begin{aligned}\|\lambda_{k+1} - \lambda^\star\|_2 &= \|[\lambda_k + \alpha g(x_{k+1})]^+ - \lambda^\star\|_2 \\ &\leq \|\lambda_k + \alpha g(x_{k+1}) - \lambda^\star\|_2 \\ &\leq \|\lambda_k\|_2 + \|\lambda^\star\|_2 + \alpha m \sigma_c.\end{aligned}$$

Next recall that when the Slater condition holds by Lemma 2.8 we have for all $\lambda \in \Lambda_\delta$ then $\|\lambda\|_2 \leq \frac{1}{\upsilon}(\eta + \delta)$ where $\eta := f(\bar{x}) - h(\lambda^\star)$ and $\bar{x}$ a Slater vector. Therefore,

$$\|\lambda_{k+1} - \lambda^\star\|_2 \leq \frac{2}{\upsilon}(\eta + \delta) + \alpha m \sigma_c.$$

From both cases, it follows that if

$$\|\lambda_1 - \lambda^\star\|_2 \leq \frac{2}{\upsilon}(\eta + \delta) + \alpha m \sigma_c$$

then $\|\lambda_k - \lambda^\star\|_2 \leq \frac{2}{v}(\eta+\delta) + \alpha m \sigma_c$ for all $k \geq 1$. Using this observation and the fact that

$$\|\lambda_1 - \lambda^\star\|_2 \geq |\|\lambda_1\|_2 - \|\lambda^\star\|_2| \geq \|\lambda_1\|_2 - \|\lambda^\star\|_2$$

we obtain that when $\|\lambda_1\|_2 \leq \frac{3}{v}(\eta+\delta)+\alpha m\sigma_c$ then $\|\lambda_k\|_2 \leq \frac{3}{v}(\eta+\delta)+\alpha m \sigma_c$ for all $k \geq 1$. That is, if we choose $\lambda_1(j) \leq \frac{3}{v}(\eta+\delta)+\alpha m\sigma_c \leq \lambda^\circ$ then $\lambda_k(j) \leq \lambda^\circ$ for all $j = 1, \ldots, m$, $k \geq 1$ and so updates $[\lambda_k + g(x_{k+1})]^+$ and $[\lambda_k + g(x_{k+1})]^{[0,\lambda^\circ]}$ are interchangeable as claimed.

Now we proceed to prove the upper and lower bounds in (2.23). For the lower bound see first that

$$\begin{aligned}
\|\lambda_{k+1}\|_2^2 &= \|[\lambda_k + \alpha g(x_{k+1})]^+\|_2^2 \\
&\leq \|\lambda_k + \alpha g(x_{k+1})\|_2^2 \\
&= \|\lambda_k\|_2^2 + \alpha^2 \|g(x_{k+1})\|_2^2 + 2\alpha \lambda_k^T g(x_{k+1}) \\
&\leq \|\lambda_k\|_2^2 + \alpha^2 m \sigma_c^2 + 2\alpha\lambda_k^T g(x_{k+1})
\end{aligned}$$

Rearranging terms and applying the latter bound recursively for $i = 1, \ldots, k$ yields

$$2\alpha \sum_{i=1}^k \lambda_i^T g(x_{i+1}) \geq \|\lambda_{k+1}\|_2^2 - \|\lambda_1\|_2^2 - \alpha^2 m\sigma_c^2 k \geq -\|\lambda_1\|_2^2 - \alpha^2 m \sigma_c^2 k.$$

The bound does not depend on sequence $\{x_k\}$, hence, it holds for any sequence of points in X. Fixing x_{i+1} to $\bar{x}_k$ for all $i = 1, \ldots, k$ we can write

$$2\alpha \sum_{i=1}^k \lambda_i^T g(\bar{x}_k) = 2\alpha k (\bar{\lambda}_k)^T g(\bar{x}_k)$$

Dividing by $2\alpha k$ and using the fact that $\|\lambda_1\|_2^2 \leq m\lambda^{\circ 2}$ yields

$$-\frac{m\lambda^\circ}{2\alpha k} - \frac{\alpha}{2} m \sigma_c^2 \leq (\bar{\lambda}_k)^T g(\bar{x}_k).$$

For the upper bound see that $\lambda_{k+1} = [\lambda_k + \alpha g(x_{k+1})]^+ \succeq \lambda_k + \alpha g(x_{k+1})$ and so we can write

$$\alpha \sum_{i=1}^k g(x_{i+1}) \preceq \sum_{i=1}^k (\lambda_{i+1} - \lambda_i) = \lambda_{k+1} - \lambda_1 \preceq \lambda_{k+1}.$$

Next, by the convexity of g we have that

$$\frac{1}{k} \sum_{i=1}^k \alpha g(x_{i+1}) \succeq \alpha g(\bar{x}_k)$$

and so it follows that $g(\bar{x}_k) \preceq \lambda_{k+1}/(\alpha k)$. Multiplying the last equation by $\bar{\lambda}_k$ and using the fact that $0 \preceq \lambda_{k+1} \preceq \lambda^\circ \mathbf{1}$ and $0 \preceq \bar{\lambda}_k \preceq \lambda^\circ \mathbf{1}$ yields the upper bound. Finally, the constraint violation bound (2.24) follows from the fact that $g(\bar{x}_k) \preceq \lambda^{\circ 2}/(\alpha k)\mathbf{1}$. ■

Lemma 2.9 is expressed in a general form where μ may be any suitable approximation to the usual Lagrange multiplier. Evidently, the lemma also applies in the special case where $\lambda_k = \mu_k$ in which case $\sigma_0 = 0$. Note from the lemma as well that the running average $\bar{x}_k$ is asymptotically attracted to the feasible region as k increases, *i.e.* $\lim_{k\to\infty} g(\bar{x}_k) \preceq 0$

We are now in a position to present one of our main results:

THEOREM 2.3 (Constrained Optimisation). *Consider constrained convex optimisation $\mathcal{P}$ and the associated sequence of non-convex optimisations $\{\tilde{\mathcal{P}}_k\}$:*

$$y_k \in \arg\min_{y\in Y} \; L((1-\beta)x_k + \beta y, \mu_k) \tag{2.26}$$

$$x_{k+1} = (1-\beta)x_k + \beta y_k \tag{2.27}$$

$$\lambda_{k+1} = [\lambda_k + \alpha g(x_{k+1})]^{[0,\lambda^\circ]} \tag{2.28}$$

Let the Slater condition hold and suppose that $|\lambda_k(j) - \mu_k(j)| \le \alpha\sigma_0$ for all $j = 1, \ldots, m$, $k \ge 1$ with $\sigma_0 \ge 0$. Further, suppose parameters α and β are selected sufficiently small that

$$0 < \alpha \le \gamma_1\gamma\beta\epsilon/(m^2(\sigma_c^2 + 2\sigma_0\sigma_c)) \tag{2.29}$$

$$0 < \beta \le (1-\gamma)\min\{\epsilon/(\bar{\rho}_L y_\circ^2), 1\} \tag{2.30}$$

with $\epsilon > 0$, $\gamma \in (0,1)$, $\gamma_1 \in (0, 1/2)$, $y_\circ := 2\max_{y\in Y}\|y\|_2$, $\bar{\rho}_L = \rho_F + \lambda^\circ\|\rho_g\|_1$ and λ° as given in Lemma 2.9. Then, for k sufficiently large ($k \ge \bar{k}$) the sequence of solutions $\{x_k\}$ to sequence of optimisations $\{\tilde{\mathcal{P}}_k\}$ satisfies:

$$\begin{aligned} &-\frac{2m\lambda^{\circ 2}}{\alpha k} - \alpha(m\sigma_c^2/2 + m^2\sigma_0\sigma_c) - 2\epsilon \\ &\le f(\bar{x}_k) - f^\star \le 2\epsilon + \alpha(m\sigma_c^2 + m^2\sigma_0\sigma_c) + \frac{3m\lambda^{\circ 2}}{2\alpha k} \end{aligned} \tag{2.31}$$

where $\bar{x}_k := \frac{1}{k}\sum_{i=\bar{k}}^{\bar{k}+k} x_{i+1}$, $\bar{\mu}_k := \frac{1}{k}\sum_{i=\bar{k}}^{\bar{k}+k} \mu_i$ and $\sigma_c := \max_{x\in X}\|g(x)\|_\infty$.

PROOF. First of all observe that since $\lambda_{k+1}(j) = [\lambda_k(j) + \alpha g_j(x_{k+1})]^{[0,\lambda^\circ]}$ we have that $|\lambda_{k+1}(j) - \lambda_k(j)| \leq \alpha\sigma_c$ for all k. Further, since $|\lambda_k(j) - \mu_k(j)| \leq \alpha\sigma_0$ then

$$\begin{aligned}|\mu_{k+1}(j) - \mu_k(j)| &= |\mu_{k+1}(j) - \mu_k(j) + \lambda_{k+1}(j) - \lambda_{k+1}(j) + \lambda_k(j) - \lambda_k(j)|\\ &\leq |\mu_{k+1}(j) - \lambda_{k+1}(j)| + |\lambda_{k+1}(j) - \lambda_k(j)| + |\lambda_k(j) - \mu_k(j)|\\ &\leq \alpha(2\sigma_0 + \sigma_c).\end{aligned}$$

That is,

$$\|\mu_{k+1} - \mu_k\|_2 \leq \alpha m(2\sigma_0 + \sigma_c) \qquad k = 1, 2, \ldots \tag{2.32}$$

Next, observe that since $L(\cdot, \lambda_k)$ has uniform bounded curvature and

$$\begin{aligned}|L(x, \mu_{k+1}) - L(x, \mu_k)| &\leq \|\mu_{k+1} - \mu_k\|_2 \|g(x_{k+1})\|_2\\ &\leq \|\mu_{k+1} - \mu_k\|_2 m\sigma_c\\ &\leq \alpha m^2(2\sigma_0\sigma_c + \sigma_c^2)\\ &\leq \gamma_1\gamma\beta\epsilon,\end{aligned}$$

it follows by Lemma 2.6 that for k sufficiently large ($k \geq \bar{k}$) then $L(x_{k+1}, \mu_k) - h(\mu_k) \leq 2\epsilon$ and therefore by Lemma 2.7

$$-\frac{m\lambda^{\circ 2}}{\alpha k} - \frac{\alpha}{2}m\sigma_c^2 - 2\epsilon \leq L(\bar{x}_k, \bar{\mu}_k) - f^\star \leq 2\epsilon + \frac{\alpha}{2}m\sigma_c^2 + \frac{m\lambda^{\circ 2}}{\alpha k}.$$

Next, see that since

$$\begin{aligned}|L(\bar{x}_k, \bar{\lambda}_k) - L(\bar{x}_k, \bar{\mu}_k)| &= (\bar{\lambda}_k - \bar{\mu}_k)^T g(\bar{x}_k)\\ &\leq \|\bar{\lambda}_k - \bar{\mu}_k\|_2 \|g(\bar{x}_k)\|_2\\ &\leq \alpha m^2\sigma_0\sigma_c\end{aligned}$$

we have that

$$\begin{aligned}&-\frac{m\lambda^{\circ 2}}{\alpha k} - \alpha(m\sigma_c^2/2 + m^2\sigma_0\sigma_c) - 2\epsilon\\ &\quad\leq L(\bar{x}_k, \bar{\lambda}_k) - f^\star \leq 2\epsilon + \alpha(m\sigma_c^2/2 + m^2\sigma_0\sigma_c) + \frac{m\lambda^{\circ 2}}{\alpha k}.\end{aligned}$$

Finally, by using the complementary slackness bound of Lemma 2.9 the stated result follows. ■

Theorem 2.3 says that by selecting step size α and smoothing parameter β sufficiently small then the average of the solutions to the sequence of non-convex optimisations $\{\tilde{\mathcal{P}}_k\}$ can be made arbitrarily close to the solution of constrained convex optimisation $\mathcal{P}$.

2.3.2.1. *Alternative Update.* Note that, by replacing use of Theorem 2.1 by Theorem 2.2 in the proof, we can replace update (2.26) by its non-convex Frank-Wolfe alternative,

$$y_k \in \arg\min_{y \in Y} \ \partial_x L(x_k, \mu_k)^T y \tag{2.33}$$
$$= \arg\min_{y \in Y} \ (\partial f(x_k) + \mu_k^T \partial g(x_k))^T y.$$

That is, we have:

COROLLARY 2.2 (Constrained Optimisation Using Frank-Wolfe Update). *Consider the setup in Theorem 2.3 but with update (2.26) replaced by (2.33). Then, there exists a finite $\bar{k}$ such that the bound given in (2.31) holds.*

2.3.3. Generalised Update. Let $X' \subseteq \text{conv}(Y)$ be any subset of the convex hull of action set Y, including the empty set. Since

$$\min_{y \in X' \cup Y} \ L((1-\beta)x_k + \beta y, \mu_k) \leq \min_{y \in Y} \ L((1-\beta)x_k + \beta y, \mu_k),$$

we can immediately generalise update (2.26) to

$$y_k \in \arg\min_{y \in X' \cup Y} L((1-\beta)x_k + \beta y, \mu_k) \tag{2.34}$$

and Theorem 2.3 will continue to apply. Selecting X' equal to the empty set we recover (2.26) as a special case. Selecting $X' = \text{conv}(Y)$ we recover the classical convex dual subgradient update as a special case. Update (2.34) therefore naturally generalises both the classical convex dual subgradient update and non-convex update (2.26). Hence, we have the following corollary.

COROLLARY 2.3 (Constrained Optimisation Using Unified Update). *Consider the setup in Theorem 2.3 but with update (2.26) replaced by (2.34). Then, there exists a finite $\bar{k}$ such that the bound given in (2.31) holds.*

2.4. Using Queues as Approximate Multipliers

In Theorem 2.3 the only requirement on the sequence of approximate multipliers $\{\mu_k\}$ is that it remains close to the sequence of Lagrange multipliers $\{\lambda_k\}$ generated by a dual subgradient update in the sense that $|\lambda_k(j) - \mu_k(j)| \leq \alpha\sigma_0$ for all k. In this section we consider the special case where sequence $\{\mu_k\}$ additionally satisfies the following,

$$\mu_{k+1} = [\mu_k + \tilde{\delta}_k]^{[0,\lambda^\circ]} \tag{2.35}$$

with $\tilde{\delta}_k \in \mathbf{R}^m$ and $\mu_1 = \lambda_1$.

We begin by recalling the following lemma, which is a direct result of [Mey08, Proposition 3.1.2].

LEMMA 2.10. *Consider sequences* $\{\lambda_k\}$, $\{\mu_k\}$ *in* $\mathbf{R}$ *given by updates* $\lambda_{k+1} = [\lambda_k + \delta_k]^{[0,\lambda^\circ]}$ *and* $\mu_{k+1} = [\mu_k + \tilde{\delta}_k]^{[0,\lambda^\circ]}$ *where* $\delta, \tilde{\delta} \in \mathbf{R}$. *Suppose* $\lambda_1 = \mu_1$ *and* $|\sum_{i=1}^k \delta_i - \tilde{\delta}_i| \leq \epsilon$ *for all* k. *Then, for all* k *we have that* $|\lambda_k - \mu_k| \leq 2\epsilon$.

PROOF. First of all observe that $|\lambda_{k+1} - \mu_{k+1}| = |[\lambda_k + \delta_k]^{[0,\lambda^\circ]} - [\mu_k + \tilde{\delta}_k]^{[0,\lambda^\circ]}| \leq |[\lambda_k + \delta_k]^{[0,\lambda^\circ]} - [\mu_k + \tilde{\delta}_k]^+| = |[\mu_k + \tilde{\delta}_k]^+ - [\lambda_k + \delta_k]^{[0,\lambda^\circ]}| \leq |[\mu_k + \tilde{\delta}_k]^+ - [\lambda_k + \delta_k]^+|$. We now proceed to bound the RHS of the last equation. Let $\Delta_k := -\min(\lambda_k + \delta_k, 0)$, *i.e.* $\lambda_{k+1} = \lambda_k + \delta_k + \Delta_k$ so that we can write

$$\lambda_{k+1} = \lambda_1 + \sum_{i=1}^{k} (\delta_i + \Delta_i)$$

Note that when $\lambda_{k+1} > 0$ then $\Delta_k = 0$, and that when $\lambda_{k+1} = 0$ then $\sum_{i=1}^k \Delta_i = -\lambda_1 - \sum_{i=1}^k \delta_i$. Next, note that since Δ_k is nonnegative for all k by construction we have that $\sum_{i=1}^k \Delta_i$ is non-decreasing in k. Using the latter observation it follows that $\sum_{i=1}^k \Delta_i = [-\lambda_1 - \min_{1\leq j\leq k} \sum_{i=1}^j \delta_i]^+$ and therefore

$$\lambda_{k+1} = \sum_{i=1}^{k} \delta_i + \max\{\Theta_k, \lambda_1\}$$

where $\Theta_k := -\min_{1\leq j\leq k} \sum_{i=1}^j \delta_i$. Now, let $\tilde{\Theta}_k := -\min_{1\leq j\leq k} \sum_{i=1}^j \tilde{\delta}_i$ and see that

$$\begin{aligned}
|\lambda_{k+1} - \mu_{k+1}| &= \left| \sum_{i=1}^{k} \delta_i + \max\{\Theta_k, \lambda_1\} - \sum_{i=1}^{k} \tilde{\delta}_i - \max\{\tilde{\Theta}_k, \lambda_1\} \right| \\
&\leq \left| \sum_{i=1}^{k} \delta_i - \tilde{\delta}_i \right| + \left| \max\{\Theta_k, \lambda_1\} - \max\{\tilde{\Theta}_k, \lambda_1\} \right| \\
&\overset{(a)}{\leq} \left| \sum_{i=1}^{k} \delta_i - \tilde{\delta}_i \right| + \left| \tilde{\Theta}_k - \Theta_k \right| \\
&= \left| \sum_{i=1}^{k} \delta_i - \tilde{\delta}_i \right| + \left| \min_{1\leq j\leq k} \sum_{i=1}^{j} \tilde{\delta}_i - \min_{1\leq j\leq k} \sum_{i=1}^{j} \delta_i \right| \\
&= \left| \sum_{i=1}^{k} \delta_i - \tilde{\delta}_i \right| + \left| \max_{1\leq j\leq k} \sum_{i=1}^{j} -\tilde{\delta}_i - \max_{1\leq j\leq k} \sum_{i=1}^{j} -\delta_i \right| \\
&\leq \left| \sum_{i=1}^{k} \delta_i - \tilde{\delta}_i \right| + \max_{1\leq j\leq k} \left| \sum_{i=1}^{j} \delta_i - \sum_{i=1}^{j} \tilde{\delta}_i \right|
\end{aligned}$$

where (a) follows easily from enumerating the four cases. Finally, since $|\sum_{i=1}^k \delta_i - \tilde{\delta}_i| \leq \max_{i\leq j\leq k} |\sum_{i=1}^j \delta_i - \tilde{\delta}_i|$ and $|\sum_{i=1}^k \delta_i - \tilde{\delta}_i| \leq \epsilon$ for all $k = 1, 2, \ldots$ the result follows. ∎

Applying Lemma 2.10 to our present context it follows that $|\lambda_k(j) - \mu_k(j)| \leq \alpha\sigma_0$ for all k (and so Theorem 2.3 holds) for every sequence $\{\tilde{\delta}_k\}$ such that $|\sum_{i=1}^k \alpha g_j(x_i) - \tilde{\delta}_i(j)| \leq \alpha\sigma_0$ for all k.

Of particular interest is the special case of optimisation $\mathcal{P}$ where the constraints are linear. That is, $g_j(x) = a(j)x - b(j)$ where $(a(j))^T \in \mathbf{R}^n$ and $b(j) \in \mathbf{R}$, $j = 1, \ldots, m$. Gathering vectors $a(j)$ together as the rows of matrix $A \in \mathbf{R}^{m\times n}$ and collecting additive terms $b(j)$ into vector $b \in \mathbf{R}^m$, the linear constraints can then be written as $Ax \preceq b$. Therefore, the dual subgradient Lagrange multiplier update in the sequence of optimisations $\{\tilde{\mathcal{P}}_k\}$ is given by

$$\lambda_{k+1} = [\lambda_k + \alpha(Ax_{k+1} - b)]^{[0,\lambda^\circ]} \tag{2.36}$$

with $x_{k+1} = (1-\beta)x_k + \beta y_k$, $y_k \in Y$. Now suppose that in (2.35) we select $\tilde{\delta}_k = \alpha(Ay_k - b_k)$ where $\{b_k\}$ is a sequence of points in $\mathbf{R}^m$. Then,

$$\mu_{k+1} = [\mu_k + \alpha(Ay_k - b_k)]^{[0,\lambda^\circ]} \tag{2.37}$$

with $\mu_1 = \lambda_1$.

Observe that in (2.37) we have replaced the continuous-valued quantity x_k with the discrete-valued quantity y_k. We have also replaced the constant b with the time-varying quantity b_k. Further, letting $Q := \mu/\alpha$ then (2.37) can be rewritten equivalently as

$$Q_{k+1} = [Q_k + Ay_k - b_k]^{[0,\lambda^\circ/\alpha]} \tag{2.38}$$

which is a discrete queue length update with increment $Ay_k - b_k$. The approximate multipliers μ are therefore scaled discrete queue occupancies.

Using Lemma 2.10 it follows immediately that Theorem 2.3 holds provided

$$|\textstyle\sum_{i=1}^k a(j)(x_i - y_i) + (b_i(j) - b(j))| \leq \alpha\sigma_0 \tag{2.39}$$

Since update $x_{k+1} = (1-\beta)x_k + \beta y_k$ yields a running average of $\{y_k\}$ we might expect that sequences $\{x_k\}$ and $\{y_k\}$ are always close and so uniform boundedness of $|\sum_{i=1}^k (b_i(j) - b(j))|$ is sufficient to ensure that (2.39) is satisfied. This is indeed the case, as established by the following theorem.

Theorem 2.4 (Queues as Approximate Multipliers). *Consider updates (2.36) and (2.37) where $\{y_k\}$ is an arbitrary sequence of points in Y, $x_{k+1} = (1-\beta)x_k + \beta y_k$, $\beta \in (0,1)$, $x_1 \in X := \mathrm{conv}(Y)$. Further, suppose that $\{b_k\}$ is a sequence of points in $\mathbf{R}^m$*

such that $|\sum_{i=1}^{k}(b_i(j)-b(j))| \le \sigma_2$ *for all* $j=1,\dots,m,\ k=1,2,\dots$ *Then,*

$$\|\mu_k - \lambda_k\|_2 \le 2m\alpha(\sigma_1/\beta + \sigma_2), \quad k=1,2,\dots$$

where $\sigma_1 := 2\max_{x\in X}\|Ax\|_\infty$.

Proof of Theorem 2.4. By Lemma 2.10 we require $|\sum_{i=1}^{k} a(j)(x_{i+1}-y_i)+b_i(j)-b(j)|$ to be uniformly bounded in order to establish the boundedness of $|\mu_k(j)-\lambda_k(j)|$ for all $k\ge 1$. However, since

$$\left|\sum_{i=1}^{k} a(j)(x_{i+1}-y_i)+b_i(j)-b(j)\right| \le \left|\sum_{i=1}^{k} a(j)(x_{i+1}-y_i)\right| + \left|\sum_{i=1}^{k} b_i(j)-b(j)\right|$$

and $|\sum_{i=1}^{k} b_i(j)-b(j)| \le \sigma_2$ by assumption, it is sufficient to show that $|\sum_{i=1}^{k} a(j)(x_{i+1}-y_i)|$ is bounded. Now observe that since $x_{i+1}=(1-\beta)x_i+\beta y_i$ we have $x_{i+1}-y_i = (1-\beta)(x_i-y_i)$. That is,

$$\sum_{i=1}^{k}(x_{i+1}-y_i) = (1-\beta)\sum_{i=1}^{k}(x_i-y_i).$$

Further, since

$$\sum_{i=1}^{k}(x_i-y_i) = \sum_{i=1}^{k-1}(x_{i+1}-y_i)+(x_1-y_k) = (1-\beta)\sum_{i=1}^{k-1}(x_i-y_i)+(x_1-y_k)$$

it follows that

$$\sum_{i=1}^{k}(x_{i+1}-y_i) = (1-\beta)^2\sum_{i=1}^{k-1}(x_i-y_i)+(1-\beta)(x_1-y_k).$$

Applying the preceding argument recursively we obtain that

$$\sum_{i=1}^{k}(x_{i+1}-y_i) = (1-\beta)(x_1-y_k)+(1-\beta)^2(x_1-y_{k-1})+\cdots+(1-\beta)^k(x_1-y_1),$$

i.e.

$$\sum_{i=1}^{k}(x_{i+1}-y_i) = \sum_{i=1}^{k}(1-\beta)^{k+1-i}(x_1-y_i). \tag{2.40}$$

Using (2.40) it follows that

$$2\alpha\left|\sum_{i=1}^{k} a(j)(x_{i+1}-y_i)\right| \le 2\alpha\left|\sum_{i=1}^{k}(1-\beta)^{k+1-i}a(j)(x_1-y_i)\right|$$

$$\le 2\alpha\sigma_1\sum_{i=1}^{k}(1-\beta)^{k+1-i} \tag{2.41}$$

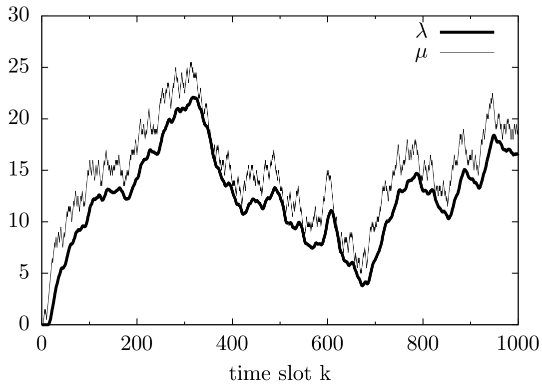


FIGURE 2.4.1. Example realisations of μ_k (thin line) and λ_k (thicker line) given by updates (2.36) and (2.37).

where $\sigma_1 := 2\max_{x\in X}\|Ax\|_\infty$. Next, observe that

$$\sum_{i=1}^{k}(1-\beta)^{k+1-i} = (1-\beta)^{k+1}\sum_{i=1}^{k}(1-\beta)^{-i}$$

and that

$$\sum_{i=1}^{k}\frac{1}{(1-\beta)^i} = \frac{1-(1-\beta)^{k+1}}{\beta(1-\beta)^k}.$$

Therefore, $\sum_{i=1}^{k}(1-\beta)^{-i} < (1-\beta)^{-k}/\beta$ and so

$$(1-\beta)^{k+1}\sum_{i=1}^{k}(1-\beta)^{-i} < \frac{(1-\beta)}{\beta} < \frac{1}{\beta}.$$

Finally, using the latter bound in (2.41) the stated result now follows. ■

Observe that the difference between λ_k and μ_k can be made arbitrarily small by selecting α small enough. The requirement that $|\sum_{i=1}^{k}(b_i(j)-b(j))| \le \sigma_2$ is satisfied when sequence $\{b_k(j)\}$ converges sufficiently fast to $b(j)$ (dividing both sides by k, the requirement is that $|\frac{1}{k}\sum_{i=1}^{k}b_i(j)-b(j)| \le \sigma_2/k$).

In the special case when $b_k(j) = b(j)$ then Theorem 2.4 is trivially satisfied. This is illustrated in Figure 2.4.1, which plots λ_k and μ_k for a simple example where $A = 1$, $b_k = b = 0.5$, $\alpha = 1$, $\beta = 0.1$ and sequence $\{x_k\}$ takes values independently and uniformly at random from set $\{0,1\}$. It can be seen that the distance between λ_k and μ_k remains uniformly bounded over time.

In summary, we have arrived at the following corollary to Theorem 2.3.

COROLLARY 2.4 (Constrained Optimisation Using Approximate Multipliers). *Consider the setup of Theorem 2.3, suppose the constraints are linear* $Ax-b \preceq 0$ *and*

$\mu_{k+1} = [\mu_k + \alpha(Ay_k - b_k)]^{[0,\lambda^\circ]}$, $b_k \in \mathbf{R}^m$. *Suppose* $|\frac{1}{k}\sum_{i=1}^k b_i(j) - b(j)| \leq \sigma_2/k$ *for all* j *and* k*. Then, the bound (2.31) in Theorem 2.3 holds with* $\sigma_0 = 2(\sigma_1/\beta + \sigma_2)$ *where* $\sigma_1 := 2\max_{x\in X}\|Ax\|_\infty$.

2.4.1. Weaker Condition for Loose Constraints. Suppose constraint j is loose at an optimum, *i.e.* $g_j(x^\star) < 0$ for $x^\star \in X^\star$. Then by complementary slackness the associated Lagrange multiplier must be zero, *i.e.* $\lambda^\star(j) = 0$, and we can select $\lambda_k(j) = \lambda^\star(j) = 0$ for all k. Since $\mu_k(j)$ is non-negative, to apply Theorem 2.3 it is enough that $\mu_k(j) \leq \alpha\sigma_0$ for $k = 1, 2, \ldots$. Assuming, for simplicity, that $\mu_1(j) = 0$, from the proof of Lemma 2.10 we have

$$\mu_k(j) = \left[\max_{1\leq l\leq k-1}\sum_{i=l}^{k-1}\alpha(a(j)y_i - b_i(j))\right]^+$$

and so a sufficient condition for $\mu_k(j) \leq \alpha\sigma_0$ is that $\max_{1\leq l\leq k-1}\sum_{i=l}^{k-1}(a(j)y_i - b(j)) - (b_i(j) - b(j)) \leq \sigma_0$ for all k. The advantage of this condition is that $-\sum_{i=l}^{k-1}(b_i(j) - b(j))$ is now not bounded below and so a wider class of sequences $\{b_i(j)\}$ is potentially admissible. The disadvantage is that to exploit this we need to know in advance that constraint j is loose at the optimum—which is something that we usually do not know in advance.

2.4.2. Queue Stability. Recall that by Lemma 2.9 sequence $\{\lambda_k\}$ in Theorem 2.3 (and respective corollaries of the theorem) is bounded for all $k \geq \bar{k}$. Therefore, since $\|\mu_k - \lambda_k\|_2$ is uniformly bounded it follows that $\{\mu_k\}$ is also bounded and therefore the associated discrete queue is stable (although the occupancy Q of the discrete queue scales with $1/\alpha$ since $Q = \mu/\alpha$).

2.4.3. Optimal Actions Depend Only on Queue Occupancy. In network resource allocation problems where the linear constraints can be identified with link queues we can use the scaled queue occupancies directly in the optimisation. That is,

$$y_k \in \arg\min_{y\in Y} L((1-\beta)x_k + \beta y, \alpha Q_k) \tag{2.42}$$

$$= \arg\min_{y\in Y} f((1-\beta)x_k + \beta y) + \alpha\beta Q_k^T Ay \tag{2.43}$$

where update (2.43) is obtained from (2.42) by retaining only the parts of $L((1-\beta)x_k + \beta y, \mu_k)$ which depend on y *i.e.* dropping constant terms which do not change the solution to the optimisation. We could also consider Corollary 2.2 and so have a Frank-Wolfe like update:

$$y_k \in \arg\min_{y \in Y} \partial_x L(x_k, \alpha Q_k)^T y \tag{2.44}$$

$$= \arg\min_{y \in Y} \partial f(x_k)^T y + \alpha Q_k^T A y \tag{2.45}$$

Importantly, note that neither (2.43) nor (2.45) involve b or b_k. Therefore, we can generate a sequence of discrete actions by simply looking at the queue occupancies at each time slot.

2.5. Max-Weight Revisited

Recall the formulation of a queueing network given at the beginning of the chapter, where matrix A defines the queue interconnection, with j'th row having a -1 at the j'th entry, 1 at entries corresponding to queues from which packets are sent to queue j, and 0 entries elsewhere. Hence, the queue occupancy evolves as

$$Q_{k+1} = [Q_k + Ay_k + b_k]^{[0,\lambda^\circ/\alpha]}.$$

As shown in Section 2.4 updates

$$y_k \in \arg\min_{y \in Y} \partial f(x_k)^T y + \alpha Q_k^T A y,$$

$$x_{k+1} = (1-\beta)x_k + \beta y_k$$

leads to x_k converging to a ball around the solution to the following convex optimisation,

$$\begin{aligned} \underset{x \in X}{\text{minimise}} \quad & f(x) \\ \text{subject to} \quad & Ax + b \preceq 0 \end{aligned}$$

where $X = \text{conv}(Y)$, $\{b_k\}$ is any sequence of points from $\mathbf{R}^m$ such that $\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^k b_i = b$ and $|(\frac{1}{k}\sum_{i=1}^k b_i(j)) - b(j)| \le \sigma_2/k$, $j = 1, \ldots, m$ for some finite $\sigma_2 > 0$.

Observe that this update is identical to the greedy primal-dual max-weight schedule once we identify utility function U with $-f$. However, we have arrived at this from a purely convex optimisation viewpoint and by elementary arguments, without recourse to more sophisticated Lyapunov drift, stochastic queueing theory, *etc.* Further, our analysis immediately generalises the max-weight analysis to allow arbitrary linear constraints rather than just the specific constraints associated with a queueing network, and beyond this to convex nonlinear constraints with bounded curvature.

In our analysis, the key role played by bounded curvature in non-convex descent is brought to the fore. This property is of course present in existing max-weight results, in the form of a requirement for continuous differentiability of the utility function, but insight into the fundamental nature of this requirement had been lacking. One immediate benefit is the resulting observation that any non-convex update with suitable descent properties can be used, and strong connections are established with the wealth of convex descent methods. For example, by Theorem 2.3 we can replace update $y_k \in \arg\min_{y\in Y}(\partial f(x_k) + AQ_k)^T y$ (which is now seen to be a variant of the classical Frank-Wolfe update) with the direct Lagrangian update

$$y_k \in \arg\min_{y\in Y} f(x_k + \beta(y - x_k)) + \beta Q_k^T A y$$

to obtain a new class of non-convex algorithms.

2.6. Numerical Example

Consider the convex optimisation problem

$$\begin{aligned} \underset{x\in X}{\text{minimise}} \quad & n \cdot \exp(Vx) \\ \text{subject to} \quad & b \preceq x \end{aligned}$$

where $V := \text{diag}(1, \dots, n)$, $X := \text{conv}(Y)$, $Y := \{y \in \mathbf{R}^n : x(j) \in \{0, s\}, j = 1, \dots, n\}$, $s > 0$ and $b = (s/\sum_{i=1}^n 2i)[1, \dots, n]^T$. Observe that the Slater condition holds. Consider the following sequence of non-convex optimisations for $k = 1, 2, \dots,$

$$\begin{aligned} y_k \in \arg\min_{y\in Y} \quad & n \cdot \exp(V((1-\beta)x_k + \beta y)) \\ & + \mu_k^T(b - ((1-\beta)x_k + \beta y)) \end{aligned} \tag{2.46}$$

$$x_{k+1} = (1-\beta)x_k + \beta y_k \tag{2.47}$$

$$\lambda_{k+1} = [\lambda_k + \alpha(b - x_{k+1})]^{[0,\lambda^\circ]} \tag{2.48}$$

with $x_1 = s\mathbf{1}$, $\lambda_1(j), \mu_1(j) = 0$, $j = 1, \dots, m$ and parameters α and β are selected as indicated in (2.29) and (2.30), with parameters $n = 3$, $s = 1/\sqrt{n\bar{\rho}_L}$, $\bar{\rho}_L = 0.6$, $\gamma = 0.5$, $\lambda^\circ = 0.7$, $\sigma_c = 0.6211$ and $y_\circ = s\sqrt{n}$.

Convergence into 2ϵ-ball in finite time. To begin with, suppose $\mu = \lambda$. For $\epsilon = 0.05$ (so $\alpha = 7.29 \cdot 10^{-5}$). Figure 2.6.1 plots the convergence of $L(x_{k+1}, \lambda_k)$ into an 2ϵ-ball around $f^\star$. It can be seen that this convergence occurs within finite time, $\bar{k} = 81$ and that $L(x_{k+1}, \lambda_k)$ then stays within this ball at times $k \geq \bar{k}$.

Upper and lower bounds from Theorem 2.3. Now suppose that $\mu(j) = \lambda(j) + \alpha Y_k \sigma_0$ where Y_k is uniformly randomly distributed between -1 and 1. For $\sigma_0 \in \{0, 1, 4\}$ (so $\alpha \in \{7.29 \cdot 10^{-5}, 1.85 \cdot 10^{-5}, 5.74 \cdot 10^{-6}\}$), Figure 2.6.2 plots $f(\bar{x}_k)$ and the upper and

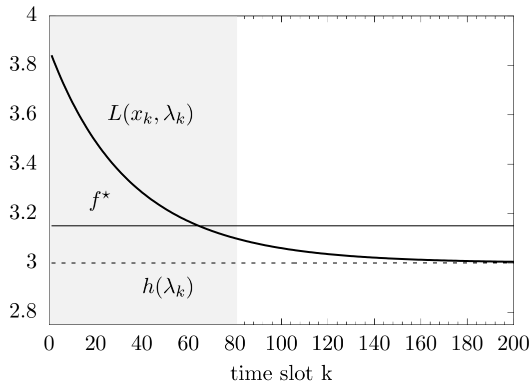
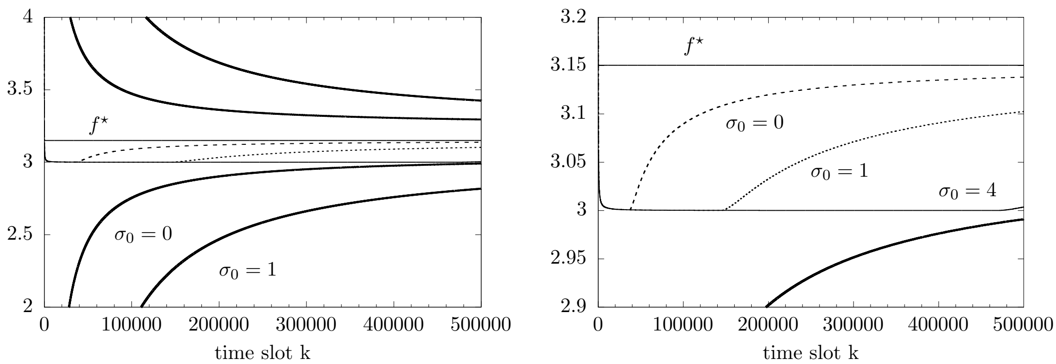


FIGURE 2.6.1. Illustrating the convergence of $L(x_{k+1}, \lambda_k)$ to a ball around $h(\lambda_k)$ for $\epsilon = 0.05$ and $\sigma_0 = 0$. Shaded area ($k < 81$) indicates that $L(x_{k+1}, \lambda_k) - h(\lambda_k) > 2\epsilon$.



(A) (B) Detail of 2.6.2a.

FIGURE 2.6.2. Illustrating the convergence of $f(\bar{x}_k)$ to a ball around $f^\star$ (straight line) of Example 2.6 when $\epsilon = 0.05$ and $\sigma_0 \in \{0, 1, 4\}$. Dashed lines indicate $f(\bar{x}_k)$ with $\bar{k} = 81$ while thick lines indicate upper and lower bounds of Theorem 2.3.

lower bounds from Theorem 2.3 vs k. Figure 2.6.2b shows detail from Figure 2.6.2. It can be seen that, as expected, $f(\bar{x}_k)$ is indeed upper and lower bounded by the values from Theorem 2.3. It can also be seen that the upper and lower bounds are not tight, but they are not excessively loose either.

Violation of upper bound. Let $\mu_k(j) = [\lambda(j)_k + \alpha e^k e^{-10^5}]^{[0,\lambda^\circ]}$. With this choice the difference between $\lambda_k(j)$ and $\mu_k(j)$ is uniformly bounded by $\alpha\sigma_0$ with $\sigma_0 = 1$ for $k \leq 10^5$ but after that increases exponentially with k. Figure 2.6.3 plots $f(\bar{x}_k)$ and the upper and lower bounds from Theorem 2.3 when parameter α is selected according to Theorem 2.3 assuming $\sigma_0 = 1$. It can be seen that the upper and lower bounds hold for $k \leq 10^5$, but as the difference between multipliers increases $f(\bar{x}_k)$ is not attracted to $f^\star$ and it ends up violating the bounds.

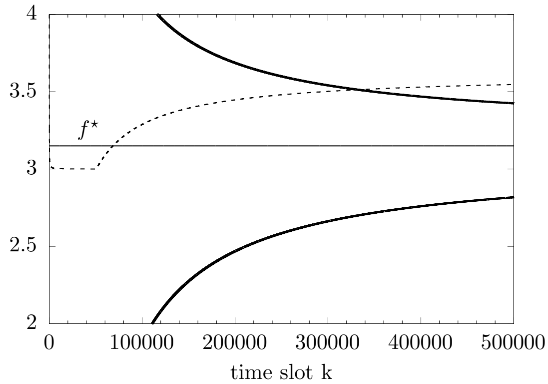


FIGURE 2.6.3. Illustrating the violation of the bounds of Theorem 2.3 when $\mu_k(j) = [\lambda_k(j) + \alpha e^k e^{-10^5}]^{[0,\lambda^\circ]}$. Dashed line indicates $f(\bar{x}_k)$, $\bar{k} = 81$, while thicker lines indicate upper and lower bounds around $f^\star$ (straight line).

CHAPTER 3

Dual Subgradient Methods with Perturbations

In this chapter we aim to step back from the non-convex descent approach presented in Chapter 2, and revisit the essentials required for convergence in dual subgradient methods. Primarily, our motivation is to build a framework for addressing network scheduling problems that can handle discrete actions and general stochastic processes, but also in developing the intuition behind the convergence of the iterative method. For example, why averages are important for recovering approximate primal solutions, and how different types of perturbations affect convergence. We start by presenting the classic subgradient method for the Lagrange dual problem, including ϵ_k-subgradients and stochastic updates; and then put these two concepts together to develop a framework based on elementary perturbations. The main novelty of the framework is to see stochastic updates as solving a convex optimisation problem where the constraints are partially unknown, and in establishing the statistical properties that perturbations should have in order to obtain different types of convergence.

3.1. Dual Subgradient Methods

We start by recalling our problem setup. We have convex optimisation problem $\mathcal{P}$

$$\begin{aligned} &\underset{x\in X}{\text{minimise}} && f(x) \\ &\text{subject to} && g_j(x) \le 0 \qquad j = 1,\dots,m \end{aligned} \tag{3.1}$$

where $f, g_j : X \to \mathbf{R}$ are convex functions, and $X \subseteq \mathbf{R}^n$ is a convex set. Now we do not assume that the objective function and constraints have bounded curvature, and X does not need to be the convex hull of an action set Y. As in the previous chapter, an essential assumption is that the relative interior of $X_0 := \{x \in X \mid g_j(x) \le 0, j = 1,\dots,m\}$ is non-empty, *i.e.* the Slater condition is satisfied and strong duality holds. Recall we define the Lagrange dual function as

$$h(\lambda) = \inf_{x\in X} L(x,\lambda) = \inf_{x\in X} \left\{ f(x) + \lambda^T g(x) \right\},$$

where $g = [g_1, \dots, g_m]^T$, and $\lambda \in \mathbf{R}^m_+$ is a vector of Lagrange multipliers. Since h is concave we can cast the following (dual) maximisation problem $\mathcal{D}$

$$\underset{\lambda \succeq 0}{\text{maximise}} \quad h(\lambda)$$

Problem $\mathcal{D}$ is an unconstrained maximisation problem that can be solved using the subgradient method, which needs to assume very little about the objective function and constraints. Namely, only that they are convex and closed, and does not require stronger assumptions such as differentiability or strict convexity. In Chapter 2, one of the underlying motivations for using a dual subgradient method was that the update of the dual variables resembled a queue update in communications networks. However, since we have now stepped back from specific applications, one might reasonably ask why we are interested in formulating the Lagrange dual problem instead of solving the primal problem $\mathcal{P}$ directly, *e.g.* using projected subgradient descent or similar methods [BT03]. Our motivation for considering the Lagrange dual problem in this chapter is that with the dual subgradient methods we can generate a sequence $\{x_k\}$ that converges to $x^\star$ (or a nearby point) without requiring that x_k is feasible at every iteration. As we will show in Section 3.3, this will be useful to "relax" the knowledge of set X_0.

Next we briefly review the essentials of the convergence of the dual subgradient method with constant step size. The ideas we present are not new, and can be found, for example, in the classic works of Polyak [Pol77] and Shor [Sho12]. Our interest is in building the intuition behind the dual subgradient method, which will be essential for the development of the subgradient method with perturbations framework in Section 3.3.

3.1.1. Classic Dual Subgradient Method. The subgradient method for the Lagrange dual problem consists of the following update:

$$\lambda_{k+1} = [\lambda_k + \alpha \partial h(\lambda_k)]^+, \qquad k = 1, 2, \dots$$

where $\lambda_1 \in \mathbf{R}^m_+$, $\partial h(\lambda_k)$ is the subgradient of h at point λ_k, and $\alpha > 0$ is a constant step size. A standard assumption made in the subgradient method is that the (dual) subgradients $\partial h(\lambda)$ are bounded for all $\lambda \succeq 0$, which can be ensured by making the following assumption.

ASSUMPTION 3.1. *X is bounded.*

Observe that since $\partial h(\lambda) = g(x)$ for some $x \in X$ we then have that

$$\|\partial h(\lambda)\|_2 \le \max_{x \in X} \|g(x)\|_2 := \sigma_g,$$

and σ_g is finite because g is a closed convex function (and so continuous) and X is bounded.

The basic idea behind the convergence of the dual subgradient method is that

(i) the Euclidean distance between λ_k and a vector $\lambda^\star \in \Lambda^\star := \arg\max_{\lambda \succeq 0} h(\lambda)$ decreases *monotonically* when λ_k is sufficiently "far away" from $\Lambda^\star$;

(ii) when λ_k is sufficiently close to $\Lambda^\star$, it remains in a ball around it.

Important characteristics of the dual subgradient method are that $\Lambda^\star$ is a bounded subset from $\mathbf{R}^m_+$ (by [NO09a, Lemma 1]); that the size of the ball to which λ_k converges depends on α; and λ_k will converge to an α-ball around $\Lambda^\star$ in finite time. Selecting α sufficiently small we can make the α-ball arbitrarily small. It is important to note that the monotonic convergence of λ_k to a ball around $\Lambda^\star$ does not imply that the value of the dual function improves in each iteration.[1] However, since the Lagrange dual function is Lipschitz continuous for $\lambda \succeq 0$ (from Assumption 3.1)[2], we can expect that $h(\lambda_k)$ converges also to a ball around $h(\lambda^\star)$. Next we show that this is actually the case. Let $x_k \in \arg\min_{x\in X} L(x, \lambda_k)$ and observe that

$$
\begin{aligned}
h(\lambda_k) - h(\lambda^\star) &= L(x_k, \lambda_k) - L(x^\star, \lambda^\star) \\
&\overset{\text{(a)}}{\leq} L(x^\star, \lambda_k) - L(x^\star, \lambda^\star) \\
&= (\lambda_k - \lambda^\star)^T g(x^\star) \\
&\leq \|\lambda_k - \lambda^\star\|_2 \|g(x^\star)\|_2
\end{aligned}
$$

and

$$
\begin{aligned}
h(\lambda_k) - h(\lambda^\star) &= L(x_k, \lambda_k) - L(x^\star, \lambda^\star) \\
&\overset{\text{(b)}}{\geq} L(x_k, \lambda_k) - L(x_k, \lambda^\star) \\
&= (\lambda_k - \lambda^\star)^T g(x_k) \\
&\geq -\|\lambda_k - \lambda^\star\|_2 \|g(x_k)\|_2
\end{aligned}
$$

where (a) and (b) follow from the saddle point property of the Lagrangian. Further, since $\|g(x)\|_2 \leq \sigma_g$ for all $x \in X$ by Assumption 3.1 we have

$$
|h(\lambda_k) - h(\lambda^\star)| \leq \|\lambda_k - \lambda^\star\|_2 \sigma_g, \tag{3.2}
$$

and so now it is easy to see that if difference $\|\lambda_k - \lambda^\star\|_2$ decreases then the difference $|h(\lambda_k) - h(\lambda^\star)|$ must eventually also decrease. The convergence of the dual subgradient

[1] By monotonic convergence we mean that the Euclidean distance between λ_k and $\Lambda^\star$ decreases.

[2] In fact, the Lagrange dual function is *uniformly* Lipschitz continuous [Die13, Chapter 3] with constant σ_g, *i.e.* for any $\lambda_1, \lambda_2 \succeq 0$ we have that $|h(\lambda_1) - h(\lambda_2)| \leq \|\lambda_1 - \lambda_2\|_2 \sigma_g$.

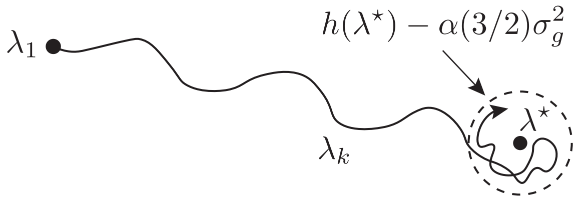


FIGURE 3.1.1. Illustrating the convergence of the subgradient method for the dual problem with constant step size.

method is schematically illustrated in Figure 3.1.1. Observe that $h(\lambda_k)$ converges to a $\alpha(3/2)\sigma_g^2$ ball around $h(\lambda^\star)$, and that $h(\lambda_k) \to h(\lambda^\star)$ as $\alpha \to 0$.

3.1.2. Computing a Subgradient of the Lagrange Dual Function. In order to use the dual subgradient method one must obtain a subgradient of the dual function, which can be obtained by first minimising $L(x, \lambda_k)$ and then evaluating $x_k \in \arg\min_{x \in X} L(x, \lambda_k)$ on the constraints, *i.e.* $\partial h(\lambda_k) = g(x_k)$. Note that minimising $L(x, \lambda_k)$ is an unconstrained convex optimisation that can be carried out with a variety of methods, and using one method or another will depend on the assumptions made on the objective function and constraints. Nevertheless, we will always assume that an $x_k \in X$ such that $L(x_k, \lambda_k) - h(\lambda_k) \le \xi$ (an ξ-approximate minimum) can be efficiently calculated for some $\xi \ge 0$.

An important observation is that an ξ-approximate minimisation can be equivalently regarded as *exactly* minimising the Lagrangian when an approximate Lagrange multiplier μ_k is used instead of λ_k. This fact is well known and is usually regarded as using ϵ_k-subgradients in dual iterative methods—see for example Bertsekas [Ber99, pp. 625]. To see this, let $x_k \in \arg\min_{x \in X} L(x, \mu_k)$ and $\mu_k = \lambda_k + \epsilon$ for some $\epsilon \in \mathbf{R}^m$, and observe that

$$
\begin{aligned}
h(\mu_k) &= L(x_k, \mu_k) \\
&= L(x_k, \lambda_k + \epsilon) \\
&= f(x_k) + (\lambda_k + \epsilon)^T g(x_k) \\
&\le f(x_k) + \lambda_k^T g(x_k) + \|\epsilon\|_2 \|g(x_k)\|_2 \\
&\le f(x_k) + \lambda_k^T g(x_k) + \|\epsilon\|_2 \sigma_g
\end{aligned}
$$

where the last inequality follows since $\sigma_g := \max_{x \in X} \|g(x)\|_2$. Hence,

$$h(\mu_k) - L(x_k, \lambda_k) \le \|\epsilon\|_2 \sigma_g. \tag{3.3}$$

We now proceed to show that $|h(\mu_k) - h(\lambda_k)|$ is bounded. Consider two cases. Case (i) $h(\mu_k) < h(\lambda_k)$. From the concavity of h we have that

$$\begin{aligned} h(\lambda_k) &\le h(\mu_k) + \partial h(\mu_k)^T(\lambda_k - \mu_k) \\ &= h(\mu_k) + \partial h(\lambda_k + \epsilon)^T \epsilon \\ &= h(\mu_k) + g(x_k)^T \epsilon, \end{aligned}$$

and therefore

$$0 \le h(\lambda_k) - h(\lambda_k + \epsilon) \le \|\epsilon\|_2 \sigma_g.$$

Case (ii) $h(\lambda_k) > h(\mu_k)$. Following the same steps than in the first case we obtain $h(\mu_k) \le h(\lambda_k) - g(x_k)^T \epsilon$, and therefore

$$0 \le h(\mu_k) - h(\lambda_k) \le \|g(x_k)\|_2 \|\epsilon\|_2 \le \|\epsilon\|_2 \sigma_g.$$

Combining both cases yields

$$|h(\lambda_k) - h(\mu_k)| \le \|\epsilon\|_2 \sigma_g,$$

and using (3.3) we finally obtain

$$0 \le L(x_k, \lambda_k) - h(\lambda_k) \le 2\|\epsilon\|_2 \sigma_\delta := \xi,$$

where the lower bound follows immediately since $h(\lambda_k) \le L(x, \lambda_k)$ for all $x \in X$. Hence, the error obtained by selecting x_k to minimise $L(x, \mu_k)$ is proportional to the difference between λ_k and μ_k. We could also consider the case where $L(x, \mu_k)$ is approximately minimised, but for the sake of clarity and to streamline notation, we will always assume that $L(x, \mu_k)$ is exactly minimised and that the errors are captured in the approximate Lagrange multipliers.

One of the interesting properties of using approximate Lagrange multipliers to compute a subgradient of the dual function, is that the ball around $\Lambda^\star$ to which λ_k converges cannot be made arbitrarily small by selecting parameter α small since

$$|h(\lambda_k) - h(\lambda^\star)| \le \|\lambda_k - \lambda^\star\|_2 \sigma_g + 2\|\epsilon\|_2 \sigma_g, \tag{3.4}$$

and therefore decreasing $\|\lambda_k - \lambda^\star\|_2$ is not sufficient to have $h(\lambda_k) \to h(\lambda^\star)$ when $\alpha \to 0$. Figure 3.1.2 schematically shows the convergence of $h(\lambda_k)$ to a ball around $h(\lambda^\star)$ when the subgradient method is computed using an approximate Lagrange multiplier. Compare Figures 3.1.1 with 3.1.2, and observe that the ball to which $h(\lambda_k)$ converges depends now

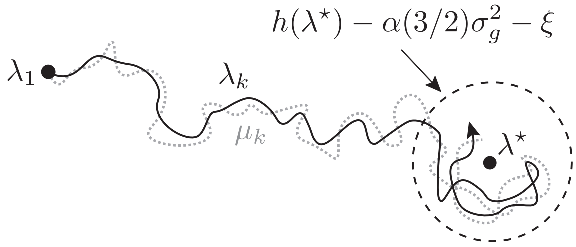


FIGURE 3.1.2. Illustrating the convergence of the subgradient method for the dual problem with constant step size when the subgradients are obtained using nearby (dual) points or approximate Lagrange multipliers. Observe from the figure that the approximate Lagrange multiplier μ_k (dashed line) is always close to the Lagrange multiplier λ_k (solid line).

on parameter $\xi = 2\|\epsilon\|_2\sigma_g$. Note also from the figure that the Lagrange multiplier and the approximate Lagrange multiplier stay (ϵ) close for all k.

3.1.3. Stochastic Dual Subgradient Methods. The subgradient method for the dual problem can be extended to consider stochastic subgradient updates, *i.e.*

$$\lambda_{k+1} = [\lambda_k + \alpha\partial\tilde{h}(\lambda_k)]^+ \tag{3.5}$$

where $\tilde{h}(\lambda_k)$ is a random variable with $\mathbf{E}(\partial\tilde{h}(\lambda_k)) = \partial h(\lambda_k)$. For example, we could have $\partial\tilde{h}(\lambda) = \partial h(\lambda_k) + N_k$ where N_k is a random variable with $\mathbf{E}(N_k) = 0$ for all k. Stochastic subgradient methods have usually been treated in unconstrained convex optimisation, and are usually thought to arise as unbiased errors when computing the subgradient of the objective function. Here, we apply the same concepts, but to the Lagrange dual problem in constrained convex optimisation.

The convergence in the stochastic dual subgradient method is (usually) not intuitive, especially when random variable N_k has unbounded support, *e.g.* N_k is normally distributed. In order to show the convergence of the stochastic subgradient method one needs to show that the *expected* difference

$$\mathbf{E}(\|\lambda_{k+1} - \lambda^\star\|_2^2 - \|\lambda_k - \lambda^\star\|_2^2)$$

decreases monotonically until $\mathbf{E}(\lambda_k)$ converges to an α-ball around a vector $\lambda^\star \in \Lambda^\star$. Figure 3.1.3 schematically shows the convergence of the stochastic dual subgradient method with approximate Lagrange multipliers. Observe that now $\mathbf{E}(\lambda_k)$ converges monotonically to a ball around $\lambda^\star$, and that the size of the ball depends on the variance σ_δ^2 of the noise in the dual subgradient, *i.e.* the noise in the stochastic subgradient method must have finite variance for the ball to have finite radius.

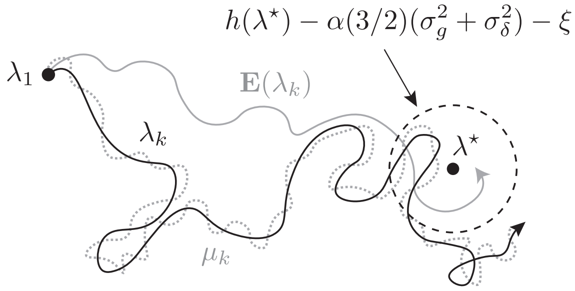


FIGURE 3.1.3. Illustrating the convergence of the stochastic subgradient method with approximate Lagrange multipliers. In contrast to the deterministic case (see Figure 3.1.2), we now have that $\mathbf{E}(\lambda_k)$ converges to a ball around $\lambda^\star$.

In general, in the stochastic subgradient method we cannot say anything about a particular value of λ_k since this will depend on the sample path of the noise in the dual variable update. However, since its increments are bounded and controlled by parameter α, one could expect λ_k to be close to its expected value, *i.e.* the sequence of Lagrange multipliers generated by the deterministic subgradient method. We illustrate this in the following example.

EXAMPLE 3.1. *We compare the convergence of the Lagrange multipliers in the deterministic and stochastic dual subgradient methods. Consider optimisation problem*

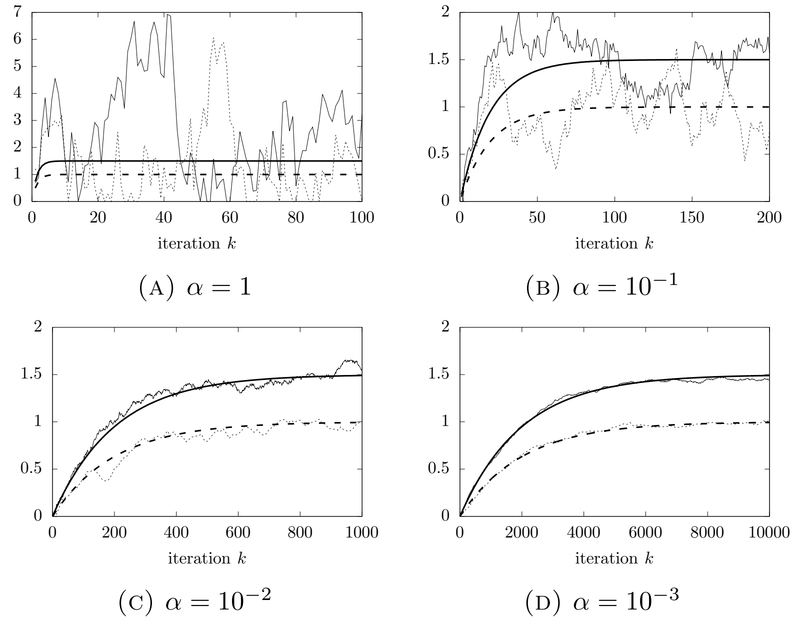
$$\begin{aligned} \underset{x \in X}{\text{minimise}} \quad & \|x - \mathbf{1}\|_2^2 \\ \text{subject to} \quad & g(x) := (x + b) \preceq 0 \end{aligned}$$

where $X = \{x \in \mathbf{R}^2 \mid 0 \preceq x \preceq \mathbf{1}\}$ *and* $b = [-1/4, -1/2]^T$, *and subgradient method updates*

$$\begin{aligned} x_k &\in \arg\min_{x \in X} \{\|x - \mathbf{1}\|_2^2 + \lambda_k^T (x + b)\} \\ \lambda_{k+1} &= [\lambda_k + \alpha(g(x_k) + N_k)]^+ \end{aligned}$$

with $N_k = 0$ *for all* k *in the deterministic case, and* $N_k(j) \sim \mathcal{N}(0, 1)$ *in the stochastic case. We run a simulation with* $\alpha \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$ *and show the convergence of the dual variables in Figure 3.1.4. Observe from the figure that with stochastic noise, when* $\alpha = 1$ *we can hardly tell that* λ_k *converges to a ball around* $\lambda^\star$*; but as the step size decreases, then the Lagrange multipliers in the stochastic subgradient method starts to mimic the behaviour of their deterministic counterpart.*

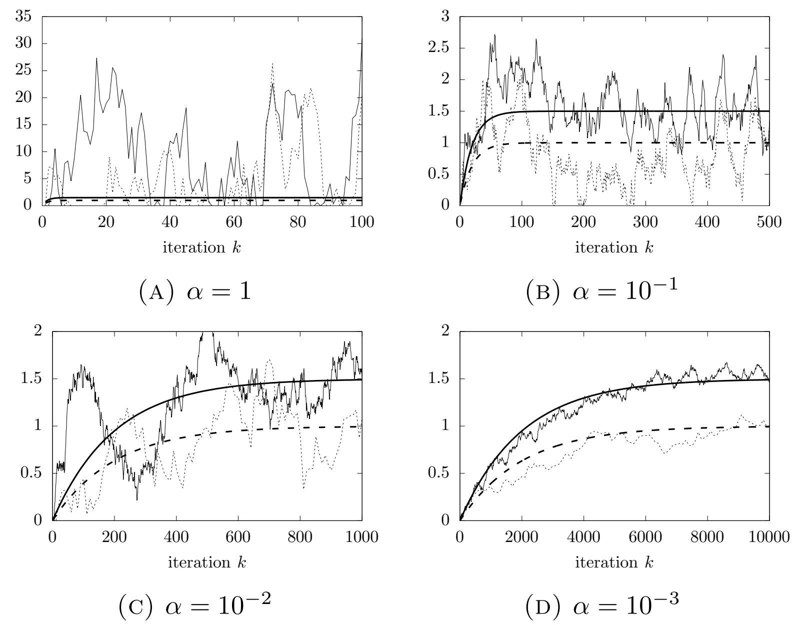
In order to show that the variance of the noise affects the size of the ball, we run the simulation again with $N_k \sim \mathcal{N}(0, 5)$. *Observe from Figure 3.1.5 that now the larger variance of* N_k *increases the distance between* $\mathbf{E}(\lambda_k)$ *and* λ_k *for all step sizes.*



(A) $\alpha = 1$ (B) $\alpha = 10^{-1}$

(C) $\alpha = 10^{-2}$ (D) $\alpha = 10^{-3}$

FIGURE 3.1.4. Illustrating the convergence of the Lagrange multipliers to a ball around $\lambda^\star = [3/2, 1]^T$ for the deterministic and stochastic dual subgradient methods, and different step sizes α. The noise in the dual variable update is normally distributed with mean 0 and variance 1. Thicker and thinner lines correspond, respectively, to the deterministic and stochastic subgradient methods. Lagrange multiplier $\lambda_k(1)$ is indicated using a solid line, and $\lambda_k(2)$ with a dashed line.



(A) $\alpha = 1$ (B) $\alpha = 10^{-1}$

(C) $\alpha = 10^{-2}$ (D) $\alpha = 10^{-3}$

FIGURE 3.1.5. As in Figure 3.1.4, but now the noise in the dual variable update is normally distributed with mean 0 and variance 5.

3.2. Bounded Lagrange Multipliers and Feasible Solutions

A primal solution can always be recovered in the dual subgradient method by minimising $L(\cdot, \lambda)$, but there is no guarantee that the primal variable obtained is feasible, neither it is possible to provide bounds on the violation of the constraints. However, as shown by Nedić and Ozdaglar in [NO09a], by using a primal averaging scheme it is possible to recover approximate primal solutions that are asymptotically feasible, and to provide bounds on the optimality gap and constraints violation. The averaging scheme used in [NO09a] consists of

$$\bar{x}_k := \frac{1}{k}\sum_{i=1}^{k} x_i \tag{3.6}$$

where $x_i \in \arg\min_{x\in X} \; L(x, \lambda_i)$. The intuition behind this sort of averaging is that the Lagrange multipliers in the dual subgradient method capture the accumulated infeasibility throughout the iterations. Hence, if λ_k is bounded when $k \to \infty$ (which is the case in the subgradient method) then $\lim_{k\to\infty} \sum_{i=1}^{k} g(x_i)$ must also be bounded. This can be verified from

$$\begin{aligned}\lambda_{k+1} = [\lambda_k + \alpha g(x_k)]^+ \succeq \lambda_k + \alpha g(x_k) &\succeq \lambda_{k-1} + \alpha(g(x_{k-1}) + g(x_k)) \\ &\succeq \lambda_1 + \alpha \textstyle\sum_{i=1}^{k} g(x_i),\end{aligned}$$

and rearranging terms, dividing by αk and using the convexity of g yields

$$g(\bar{x}_k) \preceq \frac{1}{k}\sum_{i=1}^{k} g(x_i) \preceq \frac{\lambda_{k+1}}{\alpha k}.$$

Hence, if the Lagrange multipliers remain bounded, we will have $g(\bar{x}_k) \preceq 0$ when $k \to \infty$, *i.e.* $\bar{x}_k$ is asymptotically feasible.

3.3. Framework

A key observation from the subgradient method is that we can use stochastic subgradients to tackle problems that have a priori *unknown* perturbation on the constraints. That is, problem $\mathcal{P}(\delta)$

$$\begin{array}{ll}\underset{x\in X}{\text{minimise}} & f(x) \\ \text{subject to} & g(x) + \delta \preceq 0\end{array} \tag{3.7}$$

where $\delta \in \mathbf{R}^m$ is a perturbation on the constraints. Since the optimisation problem is not known a priori we cannot use standard iterative methods in constrained convex optimisation—such as interior point methods [NN94]—to solve the problem. In fact, it is actually strange to think of using any iterative method to solve a convex optimisation problem that is not known a priori. However, if perturbation δ represents the (unknown) mean of a stochastic process in a system, and this stochastic process has certain statistical properties, it is possible to use the stochastic dual subgradient method to solve problem $\mathcal{P}(\delta)$[3]. The latter can be regarded as trying to solve the kind of problems that max-weight tries to solve: resource allocation problems where the resources that need to be allocated change over time and are not known in advance. However, the meaning of the perturbations will depend on the type of problem where the subgradient method is applied. Similarly, and unlike max-weight, Lagrange multipliers or approximate multipliers do not have to be queues, and if the Lagrange multipliers are associated with equality constraints they can even be negative.

In this section we present our main results on the subgradient method with perturbations on the computation and the update of the dual variable. Because perturbation δ is not known in advance, it will be useful to parameterise all of the elements in the problem setup with δ.

3.3.1. Parameterised Problem Setup. We consider the problem of minimising a *known* convex function $f : \mathbf{R}^n \to \mathbf{R}$ subject to a set of *partially unknown* convex constraints $g_j : \mathbf{R}^n \to \mathbf{R}$, $j = 1, \ldots, m$. Unlike previous problem setups (2.11) and (3.1), here we also consider linear equality constraints, but in order to keep notation short we write linear equality constraints as two inequality constraints[4], and collect all constraints in vector $g = [g_1, \ldots, g_m]^T$. Even though problem $\mathcal{P}(\delta)$ is not known a priori, we will assume that $X_0(\delta) := \{x \in X \mid g(x) + \delta \preceq 0\}$ is non-empty, and using standard notation we define $f^\star(\delta) = \min_{x \in X_0(\delta)} f(x)$, and $x^\star(\delta)$ to be a solution of problem $\mathcal{P}(\delta)$. Naturally, we also parameterise the Lagrangian

$$L(x, \lambda, \delta) = f(x) + \lambda^T(g(x) + \delta), \tag{3.8}$$

and the dual function $h(\lambda, \delta) := \inf_{x \in X} L(x, \lambda, \delta)$, and define dual problem $\mathcal{D}(\delta)$

$$\underset{\lambda \succeq 0}{\text{maximise}} \quad h(\lambda, \delta) \tag{3.9}$$

Similarly, we let $\lambda^\star(\delta)$ be a vector in the set of dual optima $\Lambda^\star(\delta) := \arg\max_{\lambda \succeq 0} h(\lambda, \delta)$. In order to ensure that $\Lambda^\star(\delta)$ is bounded and

$$f^\star(\delta) := \inf_{x \in X_0(\delta)} f(x) = \sup_{\lambda \succeq 0} h(\lambda, \delta) =: h(\lambda^\star(\delta), \delta) \tag{3.10}$$

[3]We will "learn" δ through the iterations in the dual subgradient method.

[4]A hyperplane can always be defined as the intersection of two halfspaces [BV04, pp. 27]

we make the following assumption.

ASSUMPTION 3.2 (Slater Condition with Linear Equality Constraints). *$X_0(\delta)$ is non-empty and there exists a point $x \in X_0(\delta)$ such that all the non-linear constraints are strictly feasible.*

Next, we present a general version of the subgradient method where errors in the computation of a (dual) subgradient, and in the update of the dual variables are captured as

$$x_k \in \arg\min_{x \in X} \; L(x, \mu_k, 0) \tag{3.11}$$

$$\lambda_{k+1} = [\lambda_k + \alpha(g(x_k) + \delta_k)]^+ \tag{3.12}$$

where $\delta_k \in \mathbf{R}^m$, and $\mu_k = \lambda_k + \epsilon_k$ for some $\epsilon_k \in \mathbf{R}^m$. Note that in update (3.11) we have used a 0 instead of δ in order to emphasise that the $\arg\min_{x \in X} L(x, \lambda, \cdot)$ does not depend on δ, *i.e.* recall that

$$\begin{aligned}\arg\min_{x \in X} L(x, \lambda, \delta) &= \arg\min_{x \in X} \{f(x) + \lambda^T (g(x) + \delta)\} \\ &= \arg\min_{x \in X} \{f(x) + \lambda^T g(x)\}.\end{aligned}$$

For the subgradient method we still make the standard assumption that set X is bounded (Assumption 3.1) and therefore the dual subgradients are bounded. Note that we have that $\|\partial_\lambda h(\lambda, \delta)\| := \max_{x \in X} \|g(x) + \delta\|_2$ where δ is unknown, but since $\delta \in \mathbf{R}^m$ there exists a constant σ_g such that $\|\partial_\lambda h(\lambda, \delta)\|_2 \le \sigma_g$ for all $\lambda \succeq 0$, and that is sufficient.

3.3.2. Dual Subgradient Method with Perturbations. We are now in position to present the following lemma.

LEMMA 3.1 (Dual Subgradient Method). *Consider optimisation problem $\mathcal{P}(\delta)$ and updates (3.11) and (3.12) where $\mu_k = \lambda_k + \epsilon_k$, with $\lambda_1 \in \mathbf{R}^m_+$ and $\epsilon_k \in \mathbf{R}^m$. Suppose $\{\delta_k\}$ is a stochastic process from $\mathbf{R}^m$ such that $\lim_{k \to \infty} \frac{1}{k} \sum_{i=1}^k \delta_i = \delta$. Then,*

$$\begin{aligned}&- \frac{\|\lambda_1 - \theta\|_2^2}{2\alpha k} - \frac{1}{k} \sum_{i=1}^k \big((\lambda_i - \theta)^T (\delta_i - \delta) + 2\|\epsilon_i\|_2 \|g(x_i) + \delta\|_2 \big) \\ &- \frac{\alpha}{2k} \sum_{i=1}^k \big(\|g(x_i) + \delta\|_2^2 + \|\delta_i - \delta\|_2^2 + 2(\delta_i - \delta)^T (g(x_i) + \delta) \big) \\ &\le \frac{1}{k} \sum_{i=1}^k h(\lambda_i, \delta) - h(\theta, \delta)\end{aligned} \tag{3.13}$$

where θ is any vector from $\mathbf{R}^m_+$.

PROOF. For any vector $\theta \in \mathbf{R}_+^m$ we have

$$\begin{aligned}\|\lambda_{k+1}-\theta\|_2^2 &= \|[\lambda_k+\alpha(g(x_k)+\delta_k)]^+-\theta\|_2^2\\ &\le \|\lambda_k+\alpha(g(x_k)+\delta_k)-\theta\|_2^2\\ &= \|\lambda_k-\theta\|_2^2+\alpha^2\|g(x_k)+\delta_k\|_2^2+2\alpha(\lambda_k-\theta)^T(g(x_k)+\delta_k)\\ &= \|\lambda_k-\theta\|_2^2+\alpha^2\|g(x_k)+\delta\|_2^2+\alpha^2\|\delta_k-\delta\|_2^2\\ &\quad +2\alpha^2(\delta_k-\delta)^T(g(x_k)+\delta)+2\alpha(\lambda_k-\theta)^T(g(x_k)+\delta_k) \qquad (3.14)\end{aligned}$$

where in the last equation we have used the fact that

$$\begin{aligned}\alpha^2\|g(x_k)+\delta_k\|_2^2 &= \alpha^2\|g(x_k)+\delta_k-\delta+\delta\|_2^2\\ &= \alpha^2\|g(x_k)+\delta\|_2^2+\alpha^2\|\delta_k-\delta\|_2^2+2\alpha^2(g(x_k)+\delta)^T(\delta_k-\delta).\end{aligned}$$

Similarly, observe that

$$\begin{aligned}(\lambda_k-\theta)^T(g(x_k)+\delta_k) &= (\lambda_k-\theta)^T(g(x_k)+\delta+\delta_k-\delta)\\ &= (\lambda_k-\theta)^T(g(x_k)+\delta)+(\lambda_k-\theta)^T(\delta_k-\delta)\end{aligned}$$

and since

$$\begin{aligned}(\lambda_k-\theta)^T(g(x_k)+\delta) &= (\lambda_k-\theta)^T(g(x_k)+\delta)+f(x_k)-f(x_k)\\ &= L(x_k,\lambda_k,\delta)-L(x_k,\theta,\delta)\\ &\le L(x_k,\lambda_k,\delta)-h(\theta,\delta), \qquad (3.15)\end{aligned}$$

we have

$$\begin{aligned}\|\lambda_{k+1}-\theta\|_2^2 \le \|\lambda_k-\theta\|_2^2 &\\ &+\alpha^2\|g(x_k)+\delta\|_2^2+\alpha^2\|\delta_k-\delta\|_2^2+2\alpha^2(\delta_k-\delta)^T(g(x_k)+\delta)\\ &+2\alpha(\lambda_k-\theta)^T(\delta_k-\delta)+2\alpha(L(x_k,\lambda_k,\delta)-h(\theta,\delta)) \qquad (3.16)\end{aligned}$$

where (3.15) follows from the fact that $h(\theta) = \min_{x\in X} L(x,\theta) \le L(x_k,\theta)$. Applying the expansion recursively for $i = 1,\ldots,k$ we have

$$\begin{aligned}\|\lambda_{k+1}-\theta\|_2^2 \le \|\lambda_1-\theta\|_2^2 &\\ &+\alpha^2\sum_{i=1}^{k}\big(\|g(x_i)+\delta\|_2^2+\|\delta_i-\delta\|_2^2+2(\delta_i-\delta)^T(g(x_i)+\delta)\big)\\ &+2\alpha\sum_{i=1}^{k}\big((\lambda_i-\theta)^T(\delta_i-\delta)+L(x_i,\lambda_i,\delta)-h(\theta,\delta)\big) \qquad (3.17)\end{aligned}$$

Next, observe that since

$$
\begin{aligned}
L(x_k, \lambda_k, \delta) &= L(x_k, \lambda_k, \delta) - L(x_k, \mu_k, \delta) + L(x_k, \mu_k, \delta) \\
&\leq |L(x_k, \lambda_k, \delta) - L(x_k, \mu_k, \delta)| + L(x_k, \mu_k, \delta) \\
&= h(\mu_k, \delta) + |L(x_k, \lambda_k, \delta) - L(x_k, \mu_k, \delta)| \\
&= h(\mu_k, \delta) + |(\lambda_k - \mu_k)^T (g(x_k) + \delta)| \\
&= h(\mu_k, \delta) + |\epsilon_k^T (g(x_k) + \delta)| \\
&\leq h(\mu_k, \delta) + \|\epsilon_k\|_2 \|g(x_k) + \delta\|_2 \\
&= h(\mu_k, \delta) - h(\lambda_k, \delta) + h(\lambda_k, \delta) + \|\epsilon_k\|_2 \|g(x_k) + \delta\|_2 \\
&\leq |h(\mu_k, \delta) - h(\lambda_k, \delta)| + h(\lambda_k, \delta) + \|\epsilon_k\|_2 \|g(x_k) + \delta\|_2 \\
&\leq h(\lambda_k, \delta) + 2\|\epsilon_k\|_2 \|g(x_k) + \delta\|_2,
\end{aligned}
$$

we have that

$$L(x_k, \lambda_k, \delta) - h(\lambda_k, \delta) \leq 2\|\epsilon_k\|_2 \|g(x_k) + \delta\|_2, \tag{3.18}$$

and therefore

$$
\begin{aligned}
\|\lambda_{k+1} - \theta\|_2^2 \leq \|\lambda_1 - \theta\|_2^2 & \\
&+ \alpha^2 \sum_{i=1}^{k} \left(\|g(x_i) + \delta\|_2^2 + \|\delta_i - \delta\|_2^2 + 2(\delta_i - \delta)^T (g(x_i) + \delta) \right) \\
&+ 2\alpha \sum_{i=1}^{k} \left((\lambda_i - \theta)^T (\delta_i - \delta) + 2\|\epsilon_i\|_2 \|g(x_i) + \delta\|_2 \right) \\
&+ 2\alpha \sum_{i=1}^{k} (h(\lambda_i, \delta) - h(\theta, \delta))
\end{aligned} \tag{3.19}
$$

Rearranging terms and dividing by $2\alpha k$ yields the stated result. ■

Lemma 3.1 establishes a lower bound on $\frac{1}{k}\sum_{i=1}^{k} h(\lambda_i, \delta) - h(\theta, \delta)$, where θ is any vector from $\mathbf{R}_+^m$, but note that when $\theta = \lambda^\star(\delta)$ then we can upper bound (3.13) by zero. We proceed to analyse the terms in the LHS of (3.13). Firstly, since by Assumption 3.1 $\lambda^\star(\delta)$ is a bounded vector then

$$\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} \tag{3.20}$$

goes to zero as $k \to \infty$. An important characteristic of (3.20) is that since it is divided by α, the convergence rate is inversely proportional to the step size used (the impact of α on the convergence rate can be observed in Figures 3.1.4 and 3.1.5 of Example 3.1). The second term on the RHS of (3.13) can be written as

$$-\alpha\left(\underbrace{\frac{1}{2k}\sum_{i=1}^{k}\|g(x_i)+\delta\|_2^2}_{\text{(a)}}+\underbrace{\frac{1}{2k}\sum_{i=1}^{k}\|\delta_i-\delta\|_2^2}_{\text{(b)}}+\underbrace{\frac{1}{k}\sum_{i=1}^{k}(\delta_i-\delta)^T(g(x_i)+\delta)}_{\text{(c)}}\right) \tag{3.21}$$

When terms (a), (b) and (c) are bounded above then (3.21) can be made arbitrarily small by selecting α sufficiently small. Term (a) is the sum of the dual subgradients. Since X is bounded we have that $\|g(x)+\delta\|_2$ is bounded by σ_g, and so (a) is bounded by $\sigma_g^2/2$. The characteristics of terms (b) and (c) depend on the characteristics of stochastic process $\{\delta_k\}$. We consider two cases. Case (i) δ_k, $k=1,2,\dots$ are uniformly bounded random variables. Then, term (b) is trivially uniformly upper bounded for all k; and since $(\delta_i-\delta)^T(g(x_k)+\delta)\le\|\delta_k-\delta\|_2\|g(x_k)+\delta\|_2\le\|\delta_k-\delta\|_2\sigma_g$ by Cauchy-Schwarz, we have that term (c) is also uniformly upper bounded. Case (ii) δ_k, $k=1,2,\dots$ are independent and have finite variance and kurtosis, but they do not necessarily have to be bounded. In this case, we can upper bound terms (b) and (c) with probability one asymptotically as $k\to\infty$ using Hoeffding's inequality [Hoe63]. Hoeffding's bound can be applied to term (b) directly, and for term (c) it is sufficient to note that

$$-\frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{m}(\delta_i(j)-\delta(j))(g_j(x_k)+\delta(j))\ge-\sum_{j=1}^{m}\left|\frac{1}{k}\sum_{i=1}^{k}(\delta_i(j)-\delta(j))\right|\sigma_g \tag{3.22}$$

where $\delta(j)$ is the j'th component of vector $\delta\in\mathbf{R}^m$.

Finally, in (3.13) we have the terms

$$-\left(\underbrace{\frac{2}{k}\sum_{i=1}^{k}\|\epsilon_i\|_2\|g(x_i)+\delta\|_2}_{\text{(d)}}+\underbrace{\frac{1}{k}\sum_{i=1}^{k}(\lambda_i-\lambda^\star(\delta))^T(\delta_i-\delta)}_{\text{(e)}}\right)$$

which do not depend on α. Since term (d) depends on sequence ϵ_k, the boundedness of the term will depend on the assumptions we make on ϵ_k. We consider three cases. Case (i) $\|\epsilon_k\|_2\le\epsilon$ for all k for some $\epsilon>0$. In this case we have that (d) can be upper bounded by $2\epsilon\sigma_g$ and therefore it is uniformly upper bounded. Case (ii) $\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^{k}\|\epsilon_i\|_2=\epsilon$. We cannot say anything about term (d) for finite k, but we will have that (d) is upper bounded by $2\epsilon\sigma_g$ when $k\to\infty$. An interesting observation is that differently from stochastic process $\{\delta_k\}$, if $\{\epsilon_k\}$ were a stochastic process, it would not need to have finite variance in order that $\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^{k}\|\epsilon_i\|_2$ exists and is finite. Case (iii) $\{\|\epsilon_k\|_2\}$ is a random variable with finite variance and mean ϵ. In this case we can use Hoeffding's inequality to give a bound on (d) with probability one asymptotically as $k \to \infty$.

Term (e) is perhaps the term for which the analysis is more delicate. In the deterministic subgradient method we have that $\delta_k = \delta$ for all k and so the term is equal to zero for all k. Observe that when (e) is nonnegative, then we can ignore the term since this would still leave a lower bound on the LHS of (3.13). However, since $\lambda^\star(\delta)$ is not known (we only know it is finite), it is not possible to determine the sign of (e), and so the term could be unbounded below when $k \to \infty$. As we will show in Theorem 3.1, it will often be useful to assume that $\{\delta_k\}$ is an ergodic process (*i.e.* $\mathbf{E}(\delta_k) = \delta$ for all k), and make use of the fact that λ_k and δ_k are independent for all k, in which case

$$\mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k}(\lambda_i - \lambda^\star(\delta))^T(\delta_i - \delta)\right) = \frac{1}{k}\sum_{i=1}^{k}\mathbf{E}(\lambda_i - \lambda^\star(\delta))^T\mathbf{E}(\delta_i - \delta) = 0$$

and the *expected* value of the lower bound in Lemma 3.1 does not depend on term (e).

In summary, from the study of the LHS of (3.13) we have obtained that $\{\delta_k\}$ needs to be uniformly bounded or have finite variance, whereas ϵ_k only needs $\lim_{k\to\infty}\sum_{i=1}^{k}\|\epsilon_i\|_2 \leq \epsilon$ for some $\epsilon > 0$ for the LHS of (3.13) to be bounded asymptotically. However, assuming that $\|\epsilon_k\|_2$ is uniformly upper bounded is useful to give bounds that are not asymptotic. We will make this assumption for the rest of the thesis.

3.3.3. Convergence. We are now in position to present one of our main theorems, which establishes the convergence to an optimum.

THEOREM 3.1. *Consider problem $\mathcal{P}(\delta)$ and updates (3.11) and (3.12) where $\mu_k = \lambda_k + \epsilon_k$ with $\{\epsilon_k\}$ a sequence of points from $\mathbf{R}^m$, and $\lambda_1 \in \mathbf{R}^m_+$. Suppose that $\|\epsilon_k\|_2 \leq \epsilon$ for all k and some $\epsilon \geq 0$, and that δ_k is an ergodic stochastic process with expected value δ and $\mathbf{E}(\|\delta_k - \delta\|_2^2) = \sigma_\delta^2$. Further, suppose Assumptions 3.1 and 3.2 hold. Then,*

(i) $\mathbf{E}(f(\bar{x}_k) - f^\star(\delta)) \leq \dfrac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} + \dfrac{\|\lambda_1\|_2^2}{2\alpha k} + 2\epsilon\sigma_g$

(ii) $\lim_{k\to\infty} |\mathbf{E}(f(\bar{x}_k) - f^\star(\delta))| \leq \dfrac{\alpha}{2}(\sigma_g^2 + \sigma_\delta^2) + 2\epsilon\sigma_g$

(iii) $\lim_{k\to\infty} \mathbf{E}\left(g(\bar{x}_k) + \delta\right) \preceq 0$

(iv) $\mathbf{E}\left(\dfrac{1}{k}\sum_{i=1}^{k}\lambda_i\right) \prec \infty \qquad k = 1, 2, \ldots$

where $\bar{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i$.

PROOF. Let $\theta = \lambda^\star(\delta)$ in Lemma 3.1. From (3.13) and (3.18) we can write

$$\begin{aligned} h(\lambda^\star(\delta),\delta) &\geq \frac{1}{k}\sum_{i=1}^{k} h(\lambda_i,\delta) \\ &\geq \frac{1}{k}\sum_{i=1}^{k}(L(x_i,\lambda_i,\delta) - 2\|\epsilon_i\|_2\|g(x_i)+\delta\|_2) \\ &= \frac{1}{k}\sum_{i=1}^{k}(f(x_i) + \lambda_i^T(g(x_i)+\delta) - 2\|\epsilon_i\|_2\|g(x_i)+\delta\|_2) \\ &\geq f(\bar{x}_k) + \frac{1}{k}\sum_{i=1}^{k}(\lambda_i^T(g(x_i)+\delta) - 2\|\epsilon_i\|_2\|g(x_i)+\delta\|_2), \end{aligned}$$

where the last equation follows from the convexity of f. Rearranging terms

$$f(\bar{x}_k) - h(\lambda^\star(\delta),\delta) \leq -\frac{1}{k}\sum_{i=1}^{k}\left(\lambda_i^T(g(x_i)+\delta) - 2\|\epsilon_i\|_2\|g(x_i)+\delta\|_2\right) \tag{3.23}$$

Now, let $\theta = 0$ in (3.14) to obtain

$$\begin{aligned} \|\lambda_{k+1}\|_2^2 \leq{}& \|\lambda_k\|_2^2 + \alpha^2\|g(x_k)+\delta\|_2^2 + \alpha^2\|\delta_k-\delta\|_2^2 \\ &+ 2\alpha^2(\delta_k-\delta)^T(g(x_k)+\delta) + 2\alpha\lambda_k^T(g(x_k)+\delta_k) \end{aligned}$$

Using the fact that $\|g(x_k)+\delta\|_2^2 \leq \sigma_g^2$ for all k and applying the latter expansion recursively

$$\begin{aligned} \|\lambda_{k+1}\|_2^2 \leq{}& \|\lambda_1\|_2^2 + \alpha^2\sigma_g^2 k + \alpha^2\sum_{i=1}^{k}\|\delta_i-\delta\|_2^2 \\ &+ 2\alpha^2\sum_{i=1}^{k}(\delta_i-\delta)^T(g(x_i)+\delta) + 2\alpha\sum_{i=1}^{k}\lambda_i^T(g(x_i)+\delta_i) \end{aligned} \tag{3.24}$$

Rearranging terms, dropping $\|\lambda_{k+1}\|_2$ since it is non-negative, and dividing by $2\alpha k$ yields

$$-\frac{1}{k}\sum_{i=1}^{k}\lambda_i^T(g(x_i)+\delta_i) \leq \frac{\|\lambda_1\|_2^2}{2\alpha k} + \frac{\alpha\sigma_g^2}{2} + \frac{\alpha}{2k}\sum_{i=1}^{k}\|\delta_i-\delta\|_2^2 + \frac{\alpha}{k}\sum_{i=1}^{k}(\delta_i-\delta)^T(g(x_i)+\delta)$$

Combining the last bound with (3.23), and using the fact that $h(\lambda^\star(\delta),\delta) = f^\star(\delta)$ (by strong duality, *c.f.* Assumption 3.2) yields

$$\begin{aligned} f(\bar{x}_k) - f^\star(\delta) \leq{}& \frac{\|\lambda_1\|_2^2}{2\alpha k} + \frac{\alpha\sigma_g^2}{2} + \frac{\alpha}{2k}\sum_{i=1}^{k}\|\delta_i-\delta\|_2^2 + \frac{\alpha}{k}\sum_{i=1}^{k}(\delta_i-\delta)^T(g(x_i)+\delta) \\ &+ \frac{1}{k}\sum_{i=1}^{k}2\|\epsilon_i\|_2\|g(x_i)+\delta\|_2 \end{aligned}$$

Taking expectations with respect to δ_i, $i = 1, 2, \ldots, k$ we have $\mathbf{E}(\|\delta_i - \delta\|_2^2) = \sigma_\delta^2$, and $\mathbf{E}((\delta_i - \delta)^T(g(x_i) + \delta)) = 0$ since by construction of (3.11) and (3.12) x_i and δ_i are independent. Therefore,

$$\mathbf{E}(f(\bar{x}_k) - f^\star(\delta)) \leq \frac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} + \frac{\|\lambda_1\|_2^2}{2\alpha k} + \frac{2}{k}\sum_{i=1}^{k} \|\epsilon_i\|_2 \sigma_g. \tag{3.25}$$

Using the fact that $\|\epsilon_k\| \leq \epsilon$ for all k in (3.25) we obtain claim (i).

We now proceed to lower bound (3.25). Taking expectations with respect to δ_i, $i = 1, 2, \ldots, k$ in Lemma 3.1, and using the fact that λ_i and δ_i are independent so $\mathbf{E}((\lambda_i - \theta)^T(\delta_i - \delta)) = 0$ and $\mathbf{E}((\delta_i - \delta)^T(g(x_i) + \delta)) = 0$, we have

$$-\frac{\|\lambda_1 - \theta\|_2^2}{2\alpha k} - \frac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} - \frac{2}{k}\sum_{i=1}^{k} \|\epsilon_i\|_2 \sigma_g \leq \mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k} h(\lambda_i, \delta) - h(\theta, \delta)\right). \tag{3.26}$$

Next, by the convexity of $-h(\cdot, \delta)$ we can write

$$\frac{1}{k}\sum_{i=1}^{k} \mathbf{E}(h(\lambda_i, \delta)) = \mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k} h(\lambda_i, \delta)\right) \leq \mathbf{E}(h(\bar{\lambda}_k), \delta)$$

and letting $\theta = \lambda^\star(\delta)$

$$-\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} - \frac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} - \frac{2}{k}\sum_{i=1}^{k} \|\epsilon_i\|_2 \sigma_g \leq \mathbf{E}\left(h(\bar{\lambda}_k, \delta) - h(\lambda^\star(\delta), \delta)\right) \leq 0, \tag{3.27}$$

where the upper bound follows from the fact that $h(\lambda^\star(\delta), \delta) = \sup_{\lambda \succeq 0} h(\lambda, \delta)$. Next, from the saddle point property of the Lagrangian

$$\begin{aligned}
\mathbf{E}(h(\bar{\lambda}_k), \delta) &\overset{(a)}{\leq} \mathbf{E}(L(\mathbf{E}(\bar{x}_k), \bar{\lambda}_k, \delta)) \\
&= \mathbf{E}(f(\mathbf{E}(\bar{x}_k)) + \mathbf{E}(\bar{\lambda}_k)^T(g(\mathbf{E}(\bar{x}_k)) + \delta))) \\
&\overset{(b)}{\leq} \mathbf{E}(f(\bar{x}_k)) + \mathbf{E}(\bar{\lambda}_k)^T \mathbf{E}(g(\bar{x}_k) + \delta),
\end{aligned}$$

where the expectation on $\bar{x}_k$ in the RHS of (a) is taken with respect to δ_i, $i = 1, \ldots, k$; and (b) follows from the convexity of f and g. Therefore,

$$\begin{aligned}
&-\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} - \frac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} - \frac{2}{k}\sum_{i=1}^{k} \|\epsilon_i\|_2 \sigma_g - \mathbf{E}(\bar{\lambda}_k)^T \mathbf{E}(g(\bar{x}_k) + \delta) \\
&\leq \mathbf{E}(f(\bar{x}_k) - f^\star(\delta)).
\end{aligned} \tag{3.28}$$

We need to show that $\mathbf{E}(\bar{\lambda}_k)^T\mathbf{E}(g(\bar{x}_k)+\delta)$ is upper bounded. Observe first that for any sequence $\{x_k\}$ from X we can write

$$\lambda_{k+1} = [\lambda_k + \alpha(g(x_k)+\delta_k)]^+ \succeq \lambda_k + \alpha(g(x_k)+\delta_k),$$

and applying the latter recursively we have that

$$\lambda_{k+1} \succeq \lambda_1 + \alpha\sum_{i=1}^{k}(g(x_i)+\delta_i).$$

Dropping λ_1 since it is non-negative, dividing by αk, and using the convexity of g it follows that

$$g(\bar{x}_k) + \frac{1}{k}\sum_{i=1}^{k}\delta_i \preceq \frac{\lambda_{k+1}}{\alpha k},$$

and taking expectations with respect to δ_i, $i=1,\dots,k$

$$\mathbf{E}(g(\bar{x}_k)+\delta)) \preceq \frac{\mathbf{E}(\lambda_{k+1})}{\alpha k}. \tag{3.29}$$

Multiplying both sides by $\mathbf{E}(\bar{\lambda}_k)$ (where the expectation is with respect to δ_i, $i=1,\dots,k$) and using Cauchy-Schwarz

$$\mathbf{E}(\bar{\lambda}_k)^T\mathbf{E}(g(\bar{x}_k)+\delta) \leq \frac{\mathbf{E}(\bar{\lambda}_k)^T\mathbf{E}(\lambda_{k+1})}{\alpha k} \leq \frac{\|\mathbf{E}(\bar{\lambda}_k)\|_2\|\mathbf{E}(\lambda_{k+1})\|_2}{\alpha k}. \tag{3.30}$$

We proceed to show that $\|\mathbf{E}(\bar{\lambda}_k)\|_2$ is bounded using Lemma 6 in [VL16]. This lemma says that for any $\chi \geq 0$ then $\mathcal{Q}_\chi := \{\lambda \succeq 0 \mid h(\lambda,\delta) \geq h(\lambda^\star(\delta),\delta) - \chi\}$ is a bounded set. Further, for any $\lambda \in \mathcal{Q}_\chi$ we have that $\|\lambda\|_2 \leq \frac{1}{\upsilon}(f(\hat{x}) - h(\lambda^\star(\delta),\delta)+\chi)$ where $\hat{x}$ is a Slater point, and $\upsilon > 0$ a constant that does not depend on χ. Now, observe that since $\mathbf{E}(h(\bar{\lambda}_k,\delta)) \leq h(\mathbf{E}(\bar{\lambda}_k),\delta)$, from (3.27) we can write

$$-\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} - \frac{\alpha(\sigma_g^2+\sigma_\delta^2)}{2} - \frac{2}{k}\sum_{i=1}^{k}\|\epsilon_i\|_2\sigma_g \leq h(\mathbf{E}(\bar{\lambda}_k),\delta) - h(\lambda^\star(\delta),\delta) \leq 0. \tag{3.31}$$

Hence, if we identify $-\chi$ with the LHS of (3.31) we obtain that $\|\mathbf{E}(\bar{\lambda}_k)\|_2$ is bounded.

We continue by giving a bound on $\|\mathbf{E}(\lambda_{k+1})\|_2$. Taking expectations in (3.19) with respect to δ_i, $i=1,\dots,k$, letting $\theta = \lambda^\star(\delta)$, $\|\epsilon_k\|_2 \leq \epsilon$, $\|g(x_k)+\delta\|_2 \leq \sigma_g$, and using the fact that λ_k and δ_k are independent for all k, we have

$$\begin{aligned}\mathbf{E}(\|\lambda_{k+1} - \lambda^\star(\delta)\|_2^2) \leq \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2+\sigma_\delta^2)k + 2\alpha\epsilon\sigma_g k \\ + 2\alpha\sum_{i=1}^{k}(h(\lambda_i,\delta) - h(\lambda^\star(\delta),\delta))\end{aligned}$$

Next, observe that since $(h(\lambda_i, \delta) - h(\lambda^\star(\delta), \delta)) \leq 0$ for all $i = 1, \ldots, k$ we can write

$$\mathbf{E}(\|\lambda_{k+1} - \lambda^\star(\delta)\|_2^2) \leq \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2)k + 2\alpha\epsilon\sigma_g k$$

and by using the convexity of $\|\cdot\|_2^2$

$$\|\mathbf{E}(\lambda_{k+1}) - \lambda^\star(\delta)\|_2^2 \leq \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2)k + 2\alpha\epsilon\sigma_g k.$$

That is, $\mathbf{E}(\lambda_{k+1})$ is within a ball around $\lambda^\star(\delta)$. Next, since $\|\lambda^\star(\delta)\|_2^2$ is bounded we can write

$$\begin{aligned}\|\mathbf{E}(\lambda_{k+1})\|_2^2 &\leq \|\lambda^\star(\delta)\|_2^2 + \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2)k + 2\alpha\epsilon\sigma_g k \\ &\leq (\|\lambda^\star(\delta)\|_2^2 + \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2) + 2\alpha\epsilon\sigma_g)k\end{aligned}$$

and therefore

$$\|\mathbf{E}(\lambda_{k+1})\|_2 \leq \sqrt{k}\sqrt{\|\lambda^\star(\delta)\|_2^2 + \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2) + 2\alpha\epsilon\sigma_g}. \tag{3.32}$$

Hence,

(3.33)

$$\mathbf{E}(\bar{\lambda}_k)^T\mathbf{E}(g(\bar{x}_k) + \delta) \leq \frac{\|\mathbf{E}(\bar{\lambda}_k)\|_2\sqrt{\|\lambda^\star(\delta)\|_2^2 + \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2) + 2\alpha\epsilon\sigma_g}}{\alpha\sqrt{k}}$$

and so we can use (3.28) to lower bound (3.25). Taking limits we obtain

$$\lim_{k\to\infty} |\mathbf{E}(f(\bar{x}_k) - f^\star(\delta))| \leq \frac{\alpha}{2}(\sigma_g^2 + \sigma_\delta^2) + 2\epsilon\sigma_g$$

as claimed in (ii).

Claim (iii) follows from (3.29) and (3.32), *i.e.*

$$\mathbf{E}(g(\bar{x}_k) + \delta) \leq \frac{\sqrt{\|\lambda^\star(\delta)\|_2^2 + \|\lambda_1 - \lambda^\star(\delta)\|_2^2 + \alpha^2(\sigma_g^2 + \sigma_\delta^2) + 2\alpha\epsilon\sigma_g}}{\alpha\sqrt{k}}. \tag{3.34}$$

Claim (iv) follows from (3.31) and the Lipschitz continuity of the dual function. ∎

Theorem 3.1 establishes that $\mathbf{E}(f(\bar{x}_k))$ converges to a ball around $f^\star(\delta)$, and that $\bar{x}_k$ converges asymptotically to a feasible point. An important observation is that the upper bound on $\mathbf{E}(f(\bar{x}_k) - f^\star(\delta))$ obtained in claim (i) is not asymptotic, and that if we select $\lambda_1 = 0$, then the second term in the upper bound can be dropped, *i.e.*

$$\mathbf{E}(f(\bar{x}_k) - f^\star(\delta)) \leq \frac{\alpha(\sigma_g^2 + \sigma_\delta^2)}{2} + 2\epsilon\sigma_g. \tag{3.35}$$

The idea behind the latter is that if $\lambda_1 = 0$ then update (3.11) becomes

$$x_1 \in \arg\min_{x\in X} L(x,0,0) \\ = \arg\min_{x\in X} f(x),$$

i.e. minimising f without taking into account the constraints. Note that then x_1 may be an infeasible point[5] and so $\bar{x}_k$ is attracted to $x^\star(\delta)$ from the exterior $(X \setminus X_0(\delta))$.

In Chapter 2 we restricted the stochastic processes that could be admitted in the optimisation by requiring that $\|\sum_{i=1}^k \delta_i - \delta\|_2$ be uniformly bounded. This requirement seems now too restrictive and a special case of Theorem 3.1. We summarise this with the following corollary.

Corollary 3.1. *Consider problem $\mathcal{P}(\delta)$ and updates (3.11) and (3.12). Instead of assuming that $\{\delta_k\}$ is an ergodic stochastic process with $\mathbf{E}(\delta_k) = \delta$ for all k, suppose that it is a sequence that satisfies $\|\sum_{i=1}^k(\delta_i - \delta)\|_2 \le e_1$ for all $k = 1, 2, \dots$ and some $\epsilon_1 \ge 0$. Further, suppose that $\|\lambda_k - \mu_k\|_2 \le e_2$ for all $k = 1, 2, \dots$ and some constant $e_2 \ge 0$, and that Assumptions 3.1 and 3.2 hold. Then,*

(i) $$-\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} - \frac{\alpha\sigma_g^2}{2} - 2\epsilon\sigma_g - \frac{\bar{\lambda}_k^T\lambda_{k+1}}{\alpha k} \le f(\bar{x}_k) - f^\star(\delta) \le \frac{\alpha\sigma_g^2}{2} + \frac{\|\lambda_1\|_2^2}{2\alpha k} + 2\epsilon\sigma_g$$

(ii) $$g(\bar{x}_k) + \delta \preceq \frac{\bar{\lambda}_k^T\lambda_{k+1}}{\alpha k}$$

where $\epsilon = 2\alpha e_1 + e_2$, $\bar{\lambda}_k = \frac{1}{k}\sum_{i=1}^k \lambda_i$.

Proof. We start by showing that condition $\|\sum_{i=1}^k \delta_i - \delta\|_2 \le e_1$ implies that $\|\delta_k - \delta\|_2 \le 2e_1$ for all k. Let $\gamma_k := \sum_{i=1}^k(\delta_i - \delta)$ and note that since $\|\cdot\|_\infty \le \|\cdot\|_2$ then $|\gamma_k(j)| \le e_1$ for all $j = 1, \dots, m$. Next, observe that we can write $|\gamma_k(j) + \delta_{k+1}(j) - \delta(j)| \le e_1$ and

$$\begin{aligned} |\delta_{k+1}(j) - \delta(j)| &= |\gamma_k(j) - \gamma_k(j) + \delta_{k+1}(j) - \delta(j)| \\ &= |\gamma_k(j) + \delta_{k+1}(j) - \delta(j)| + |\gamma_k(j)| \\ &\le e_1 + |\gamma_k(j)| \\ &\le 2e_1. \end{aligned}$$

Now, for all $j = 1, \dots, m$ we can write

$$\begin{aligned} \lambda_{k+1}(j) &= [\lambda_k(j) + \alpha(g_j(x_k) + \delta_k(j))]^+ \\ &= [\lambda_k(j) + \alpha(g_j(x_k) + \delta_k(j) + \delta(j) - \delta(j))]^+, \end{aligned}$$

[5] If at least one of the constraints is active at the optimum.

and observe that we either have that

$$\lambda_{k+1}(j) \geq [\lambda_k(j) + \alpha(g_j(x_k) + \delta(j))]^+ + \delta_k(j) - \delta(j)$$

or

$$\lambda_{k+1}(j) \leq [\lambda_k(j) + \alpha(g_j(x_k) + \delta(j))]^+ + \alpha(\delta_k(j) - \delta(j)).$$

Hence,

$$|\lambda_{k+1}(j) - [\lambda_k(j) + \alpha(g_j(x_k) + \delta(j)]^+| \leq \alpha|\delta_k(j) - \delta(j)|$$

and therefore

$$\|[\lambda_k + \alpha(g(x_k) + \delta)]^+ - \lambda_{k+1}\|_2 \leq \alpha\|\delta_k - \delta\|_2 \leq 2\alpha e_1. \tag{3.36}$$

That is, condition $\| \sum_{i=1}^k \delta_i - \delta\|_2 \leq e_1$ is equivalent to regarding λ_{k+1} as an approximate Lagrange multiplier that stays uniformly close to update $[\lambda_k + \alpha(g(x_k) + \delta)]^+$, *i.e.* the deterministic version of the subgradient method. Further, since $\mu_k = \lambda_k + \epsilon_k$ for some $\epsilon_k \in \mathbf{R}^m$ such that $\|\epsilon_k\|_2 \leq e_2$ for all k, we have that

$$\|[\lambda_k + \alpha(g(x_k) + \delta)]^+ - \mu_{k+1}\|_2 \leq 2\alpha e_1 + e_2. \tag{3.37}$$

That is, we could consider the deterministic version of the subgradient method with an approximate Lagrange multiplier that is $2\alpha e_1 + e_2$ close to multiplier $[\lambda_k + \alpha(g(x_k) + \delta)]^+$ generated by the deterministic subgradient method.

Using this observation, we can let $\theta = \lambda^\star(\delta)$ in Lemma 3.1 and write

$$-\frac{\|\lambda_1 - \theta\|_2^2}{2\alpha k} - 2\epsilon\sigma_g - \frac{\alpha}{2}\sigma_g^2 \leq \frac{1}{k}\sum_{i=1}^k h(\lambda_i, \delta) - h(\lambda^\star(\delta), \delta)$$

Using the same steps that in Theorem 3.1 one can show that

$$f(\bar{x}_k) - f^\star(\delta) \leq \frac{\alpha\sigma_g^2}{2} + \frac{\|\lambda_1\|_2^2}{2\alpha k} + 2\epsilon\sigma_g \tag{3.38}$$

and

$$-\frac{\|\lambda_1 - \lambda^\star(\delta)\|_2^2}{2\alpha k} - \frac{\alpha\sigma_g^2}{2} - 2\epsilon\sigma_g - \frac{\bar{\lambda}_k^T\lambda_{k+1}}{\alpha k} \leq f(\bar{x}_k) - f^\star(\delta). \tag{3.39}$$

where λ_k is bounded since it is monotonically attracted to a ball around a vector $\lambda^\star(\delta) \in \Lambda^\star(\delta)$. The boundedness of $\bar{\lambda}_k$ follows from the boundedness of λ_k. ■

Corollary 3.1 says that requirement $\| \sum_{i=1}^{k} \delta_i - \delta \|_2 \leq \epsilon_1$ is equivalent to considering the deterministic version of the dual subgradient method with approximate Lagrange multipliers. When, $\epsilon = 0$ we recover the result by Nedić and Ozdaglar in [NO09a, Proposition 3].

CHAPTER 4

Actions and Asynchronous Updates

In this chapter, we show how the perturbations ϵ_k in the optimisation framework in Chapter 3 can be used to (i) equip the dual subgradient method with discrete actions, and (ii) make asynchronous dual updates. In Chapter 2 we usually regarded a discrete action as a packet transmission, but it does not need to be always the case. To avoid confusion, we make the following definitions:

Discrete Action: A point from a finite collection of points from $Y \subset \mathbf{R}^n$.

Continuous Action: A value from $X \subseteq \text{conv}(Y)$.

Meta-Action: A finite (sub)sequence of discrete actions.

For example, if a discrete action represents transmitting a bit then a meta-action could be a packet transmission. The distinction between action and meta-action might seem unnecessary, but as we will show later, it will be very useful to model distributed asynchronous systems, and problems that have constraints on how discrete actions can be selected.

The structure of the chapter is as follows. We start by presenting the problem setup and how actions fit in the optimisation framework presented in Chapter 3. Next, in Section 4.2, we show how approximate Lagrange multipliers can be used to have asynchronous updates in the dual subgradient method, and motivate the importance of differentiating actions from meta-actions. Finally, in Section 4.3, we present the main results of the chapter: how to select actions in order to model meta-actions.

4.1. Preliminaries

4.1.1. Problem and Action Set. Consider the optimisation problem $\mathcal{P}(\delta)$ presented in Section 3.3, and suppose that $X \subseteq \text{conv}(Y)$ where Y is a finite collection of points from $\mathbf{R}^n$. Note that a point $x \in X$ can be written as the convex combination of the points in Y, and that differently from Chapter 2, now we allow X to be a subset from $\text{conv}(Y)$. This is not just made for the sake of generality. We will show in the example in Section 5.3 that it is useful for capturing the characteristics of some problems. Figure 4.1.1 shows examples of such sets.

We will usually think of set Y as a collection of points without any special structure, but sometimes Y might be the result of the cartesian product of $\mathcal{K}$ orthogonal sets Y_κ,

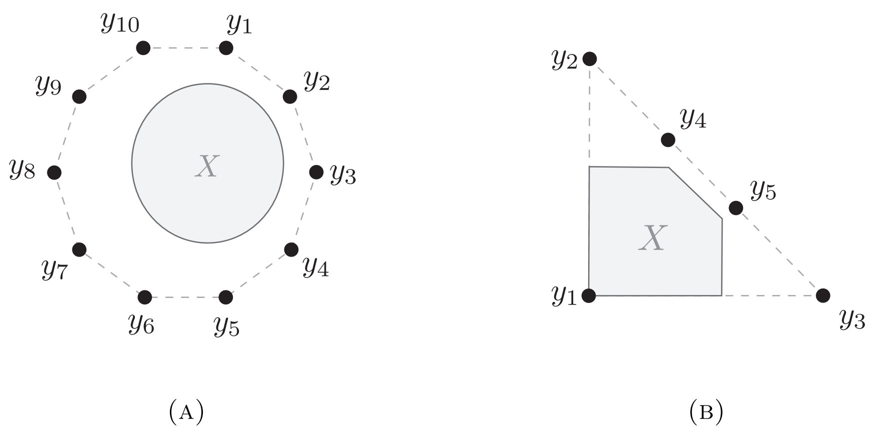


FIGURE 4.1.1. Illustrating two sets Y consisting of finite collection of points from $\mathbf{R}^2$, their respective convex hulls (dashed lines), and sets $X \subseteq \text{conv}(Y)$.

$\kappa = 1, \dots, \mathcal{K}$, *i.e.* $Y = \prod_{\kappa=1}^{\mathcal{K}} Y_\kappa$, and $n = \sum_{\kappa=1}^{\mathcal{K}} \dim(Y_\kappa)$ so that $Y \subset \mathbf{R}^n$. Similarly, convex set X can also be the result of the cartesian product of convex subsets $X_\kappa \subseteq \text{conv}(Y_\kappa)$, $\kappa = 1, \dots, \mathcal{K}$, *i.e.* $X = \prod_{\kappa=1}^{\mathcal{K}} X_\kappa$ and so we have $X \subseteq \text{conv}(Y)$.

4.1.2. Discrete Actions and Approximate Lagrange Multipliers. As in Chapter 2, we consider that approximate Lagrange multipliers have a queue-like form and are associated with linear inequality constraints. We recall the following lemma, which is a restatement of [Mey08, Proposition 3.1.2].

LEMMA 4.1 (Continuity of the Skorokhod Map). *Consider updates*

$$\lambda_{k+1} = [\lambda_k + \alpha(Ax_k + \delta_k)]^+ \tag{4.1}$$

$$\mu_{k+1} = [\mu_k + \alpha(Ay_k + \delta_k)]^+ \tag{4.2}$$

where $\lambda_1 = \mu_1 \geq 0$, $\alpha > 0$, $A \in \mathbf{R}^{m \times n}$, $\delta_k \in \mathbf{R}^m$, *and* $\{x_k\}$ *and* $\{y_k\}$ *are two sequences of points from* X *and* Y *such that* $\| \sum_{i=1}^k x_i - y_i \|_2 \leq \epsilon$. *Then,*

$$\|\lambda_k - \mu_k\|_2 \leq 2\alpha \|A\|_2 \epsilon, \qquad k = 1, 2, \dots \tag{4.3}$$

Lemma 4.1 says that if $\| \sum_{i=1}^k x_i - y_i \|_2$ is uniformly bounded then the difference $\|\lambda_k - \mu_k\|_2$ is also bounded, *i.e.* μ_k is an approximate Lagrange multiplier and so we will be able to use it in Theorem 3.1. Importantly, note that the difference between the multipliers in (4.3) depends now on α. This is important because in Theorem 3.1, the bounds on $\mathbf{E}(f(\bar{x}_k) - f^\star(\delta))$ can then be made arbitrarily small by selecting α small. Another important observation from (4.3) is that it does not depend on perturbation δ_k

since any term that appears in both the Lagrange and approximate multiplier updates gets cancelled.

When Y is the cartesian product of $\mathcal{K}$ orthogonal sets, condition $\| \sum_{i=1}^{k} x_i - y_i \|_2$ can be split in $\mathcal{K}$ different terms. To see this, observe that by using the triangle inequality we can write

$$\left\| \sum_{i=1}^{k} x_i - y_i \right\|_2 \leq \sum_{\kappa=1}^{\mathcal{K}} \left\| \sum_{i=1}^{k} x_{\kappa,i} - y_{\kappa,i} \right\|_2 , \tag{4.4}$$

where $x_{\kappa,k} \in X_\kappa$, $y_{\kappa,k} \in Y_\kappa$. Hence, if $\| \sum_{i=1}^{k} x_{\kappa,i} - y_{\kappa,i} \|_2$ is uniformly bounded for all k and $\kappa \in \{1, \ldots, \mathcal{K}\}$, then $\| \sum_{i=1}^{k} x_i - y_i \|_2$ will also be uniformly upper bounded. This will be important in terms of scalability when set Y is large (on the order of several hundred of thousands or millions of points) because the construction of sequences of discrete actions can be divided in subproblems.

4.2. Asynchronous Dual Updates

When the Lagrange dual problem allows decomposition[1] we can split it into subproblems that can be solved in a distributed but coordinated manner[2]. Decomposing the dual problem into smaller subproblems is particularly useful in large scale problems that cannot be solved in a centralised manner, but also in problems that are distributed in nature. This is the case for communication networks [Ber98], consensus [SJR16, CNS14], games [YJ10], auctions [IGHT15], *etc.*, where a set of interconnected agents solve a local subproblem, and then communicate the solution to their neighbours in order to solve a global problem. Asynchronous updates are not new and have been extensively studied by Tsitsiklis for a variety convex optimisation algorithms [Tsi84], and later by Bertsekas [BB87], Tseng [Tse91], Nedić [SN11], amongst many others. Here, we apply the totally asynchronous model in [BT89, Chapter 6] to the dual problem via the use of approximate Lagrange multipliers. The important part in this section is not to show that we can make asynchronous updates, but that we can use approximate multipliers to model the characteristics of a problem/system. As we will show later, this point motivates differentiating discrete actions from meta-actions. We start by presenting the problem setup that allows the dual problem to be decomposed in multiple subproblems.

[1]This is usually the case when the objective function is separable and the constraints are linear.
[2]See [PC06] for a tutorial on decomposition methods.

4.2.1. Dual Decomposition. Consider the following problem

$$\begin{array}{lll} \underset{x\in X}{\text{minimise}} & f(x) = \sum_{s=1}^{n} f_s(x(s)) & \\ \text{subject to} & g_j(x) = \sum_{s=1}^{n} g_{j,s}(x(s)) \leq 0 & j = 1, \ldots, m \end{array} \tag{4.5}$$

where $f_s, g_{j,s} : \mathbf{R} \to \mathbf{R}$, $s = 1, \ldots, n$, $j = 1, \ldots, m$ are convex functions, $X := \prod_{s=1}^{n} X_s$ with X_s a bounded convex subset from $\mathbf{R}$. The problem can be generalised to cases where the dimension of X_s is larger than one, but for sake of readability and simplicity of exposition, here we consider the case where X is element-wise decomposable. Note that each constraint g_j depends on all of the $x(s)$, $s = 1, \ldots, n$, and so constraints and primal variables are coupled. The Lagrangian can be written as

$$L(x, \lambda) = \sum_{s=1}^{n} L_s(x(s), \lambda) = \sum_{s=1}^{n} \left\{ f_s(x(s)) + \sum_{j=1}^{m} \lambda(j) g_{j,s}(x(s)) \right\},$$

and if we let $h_s(\lambda) = \min_{x \in X_s} L_s(x, \lambda)$ the Lagrange dual function can be expressed as the sum of n concave functions. Thus, the dual problem is

$$\underset{\lambda \succeq 0}{\text{maximise}} \quad \sum_{s=1}^{n} h_s(\lambda)$$

Importantly, a (dual) subgradient can be obtained in a distributed manner by minimising $L_s(x, \lambda)$ over X_s. We can write the dual subgradient method with approximate Lagrange multipliers associated to problem (4.5) as follows

$$\begin{aligned} x_k(1) &\in \arg \min_{x \in X_1} L_1(x, \mu_{1,k}) \\ x_k(2) &\in \arg \min_{x \in X_2} L_2(x, \mu_{2,k}) \\ &\vdots \\ x_k(n) &\in \arg \min_{x \in X_n} L_n(x, \mu_{n,k}) \\ \lambda_{k+1} &= [\lambda_k + \alpha g(x_k)]^+ \end{aligned}$$

where $\alpha > 0$, $g = [g_1, \ldots, g_m]^T$, $x_k = [x_k(1), \ldots, x_k(n)]^T$, and note we have used a different approximate Lagrange multiplier $\mu_{s,k}$, $s = 1, \ldots, n$ for each $x_k(s)$. Observe that even though $x_k(s)$ can be obtained in a distributed manner, the Lagrange multiplier update requires a full vector $x_k = [x_k(1), \ldots, x_k(n)]^T$ in order to obtain an ϵ_k-subgradient $g(x_k)$. That is, if we associate $x_k(s)$ with an agent and its value with a continuous action[3], we have that each agent must make exactly one continuous action every time the dual variable is updated.

[3]A value $x \in X$ is the convex combinations of the points in Y.

4.2.2. Asynchronous Updates. Approximate Lagrange multipliers can be used to capture asynchronous updates/continuous actions. To begin with, suppose that all agents have access to the exact Lagrange multiplier in the system and that the time in the system is divided in time slots of equal duration.[4] All the agents have to make an update in each time slot, but not all of them want to change their continuous action (a new update). Let $\mathcal{S}_k \subseteq \{1, \dots, n\}$ denote the subset of agents that want to make a new update in time slot k, and $\mathcal{S}'_k \subseteq \{1, \dots, n\} \setminus S$ the subsets of agents that do not want to. In order to capture asynchronous updates, we allow agents in set $\mathcal{S}'_k$ to use an approximate Lagrange multiplier at iteration k. Namely, if an agent wants to make a new update at time k it uses the true Lagrange multiplier in the system; otherwise an approximate Lagrange multiplier that is equal to the last Lagrange multiplier used to make a new update, *i.e.*

$$x_k(s) = \begin{cases} \arg\min_{x \in X_s} L_s(x, \lambda_k) & s \in \mathcal{S}_k \\ \arg\min_{x \in X_s} L_s(x, \lambda_{k-\tau_{s,k}}) & s \in \mathcal{S}'_k \end{cases} \tag{4.6}$$

where $\tau_{s,k} \in \mathbf{Z}_+$ captures the number of consecutive iterations an agent $s \in \mathcal{S}'$ has "skipped" making a new update. Note that update $x_k(s) \in \arg\min_{x \in X_s} L_s(x, \lambda_{k-\tau_{s,k}})$ is equivalent to selecting

$$x_k(s) = x_{k-1}(s),$$

which can be regarded as doing "nothing". The updates in (4.6) are just standard subgradient updates where the approximate multipliers are used to model a specific kind of behaviour, and if the difference $\|\lambda_k - \lambda_{k-\tau_{s,k}}\|_2$ is uniformly bounded for all k, then from Theorem 3.1 we have that the dual subgradient method with asynchronous updates will converge. We have the following corollary.

Corollary 4.1 (Asynchronous Convex Updates). *Consider problem (4.5) with the setup of Theorem 3.1 and updates*

$$x_k(s) \in \arg\min_{x \in X_s} L_s(x, \mu_{s,k}) \qquad s = 1, \dots, n$$
$$\lambda_{k+1} = [\lambda_k + \alpha g(x_k)]^+$$

where $x_k = [x_k(1), \dots, x_k(n)]^T$ *and* $\mu_{s,k} = \lambda_{k-\tau_{s,k}}$, $s = 1, \dots, n$ *for some* $\tau_{s,k} \in \mathbf{Z}_+$. *Suppose* $\tau_{s,k+t} = 0$ *for all* $s \in \{1, \dots, n\}$ *and* k *and some* $t \in \{1, \dots, \bar{\tau}\}$ *and* $\bar{\tau} \in \mathbf{Z}_+$. *Then, the bounds in Theorem 3.1 hold with* $\epsilon := \alpha\sqrt{m}\bar{\tau}\sigma_g$ *where* $\sigma_g := \max_{x \in X} \|g(x)\|_2$.

[4]A slot is equivalent to an iteration in the dual subgradient method.

PROOF. To start, note that the requirement that $\tau_{s,k+t} = 0$ means that the time between two any new updates is upper bounded by $\bar{\tau}$. To prove the corollary it is sufficient to show that $\|\lambda_{k-\tau_{s,k}} - \lambda_{k+1}\|_2 \leq \alpha\sqrt{m}\bar{\tau}\sigma_g$ for all k. Observe that

$$\begin{aligned}\lambda_{k+1} = [\lambda_k + g(x_k)]^+ \succeq \lambda_k + g(x_k) \succeq \lambda_{k-\tau_{s,k}} + \alpha \sum_{i=k-\tau_{s,k}}^{k} g(x_i) \\ \succeq \lambda_{k-\tau_{s,k}} - \alpha\mathbf{1} \sum_{i=k-\tau_{s,k}}^{k} \sigma_g\end{aligned}$$

where the last equation follows since $\|g(x)\|_2 \leq \sigma_g$ for all $x \in X$ and $\|\cdot\|_\infty \leq \|\cdot\|_2$. Similarly, we can write $\lambda_{k+1} \preceq \lambda_{k-\tau_{s,k}} + \alpha\mathbf{1}\sum_{i=k-\tau_{s,k}}^{k} \sigma_g$, and therefore

$$\begin{aligned}\|\lambda_{k-\tau_{s,k}} - \lambda_{k+1}\|_\infty &\leq \alpha \sum_{i=k-\tau_{s,k}}^{k} \sigma_g \\ &\leq \alpha\tau_{s,k}\sigma_g \\ &\leq \alpha\bar{\tau}\sigma_g\end{aligned}$$

Finally, since for a vector $\lambda \in \mathbf{R}^m$ we have that $\|\lambda\|_2 \leq \sqrt{m}\|\lambda\|_\infty$, the stated result now follows. ∎

Corollary 4.1 can be generalised in many ways, for example, we could consider that the agents do not have access to the true Lagrange multiplier in the system but to an approximate Lagrange multiplier when making an update; stochastic dual variables updates; perturbations on the constraints, *etc.* However, what is really important is that we have regarded making asynchronous updates as choosing an approximate Lagrange multiplier in the iterations of the dual subgradient method.

A simple example of when the updates in (4.6) are useful is the following. Suppose that we have a linear programme and that $X_s = [0, 1]$, *i.e.* $x_k(s)$ takes values $\{0, 1\}$ in each iteration k. If selecting $x_k(s) = 1$ means transmitting a bit from agent s, then agents/nodes in a network could use approximate Lagrange multipliers as indicated in (4.6) to model asynchronous packet transmissions. However, this approach to modelling discrete actions using continuous actions only works in special cases, and only allows to choose the action selected in the previous iteration. For more general cases we will need to use the results in the next section.

4.3. Sequences of Non-Convex Actions

Here, we present how to construct sequences of discrete actions in order to model meta-actions that capture the behaviour of a system. While the characteristics of meta-actions depend on the details of the problem, we will focus on the construction of sequences of discrete actions that provide some flexibility in the order in which actions can be selected. All the results presented in this section can be applied when Y is decomposable, but to streamline notation, we will work with sequences $\{x_k\}$ and $\{y_k\}$ from X and Y.

In short, we want to show that for an arbitrary sequence of points $\{x_k\}$ from X, we can select a sequence $\{y_k\}$ of points from Y that keeps difference $\| \sum_{i=1}^k x_i - y_i\|_2$ uniformly bounded, since by Lemma 4.1 this ensures that the exact and approximate Lagrange multipliers (λ_k and μ_k) are close. In general, it is not straightforward to show this directly, and it will be more convenient to apply a transformation and write each point $x \in X$ as the convex combination of points from Y. Collect the points in Y as columns in matrix W and define

$$\begin{aligned} E :=& \{v_1, \dots, v_{|Y|}\}, \\ U :=& \mathrm{conv}(E) = \{u \in [0,1]^{|Y|} \mid \mathbf{1}^T u = 1\}, \end{aligned}$$

where v_j is an $|Y|$-dimensional standard basis vector, *i.e.* all elements of vector v_j are equal to 0 except the j'th element that is equal to 1. Next, note that since we can always write a vector $x_i \in X$ as the convex combination of points from Y, there must exist at least one vector $u_i \in U$ such that $x_i = Wu_i$.[5] Similarly, there exists a vector $e_i \in E$ such that $y_i = We_i$. Hence,

$$\left\| \sum_{i=1}^k x_i - y_i \right\|_2 = \left\| \sum_{i=1}^k W u_i - W e_i \right\|_2 \leq \|W\|_2 \left\| \sum_{i=1}^k u_i - e_i \right\|_2, \tag{4.7}$$

and therefore showing that $\| \sum_{i=1}^k u_i - e_i\|_2$ is uniformly bounded is sufficient to establish the boundedness of $\| \sum_{i=1}^k x_i - y_i\|_2$.

4.3.1. Blocks of Discrete Actions. Now, we present a class of sequences that works with blocks or groups of discrete actions. The special characteristic of these sequences is that the actions in the blocks can be rearranged/shuffled in order to match the meta-actions of a system. Consider the following lemma.

LEMMA 4.2. *Let E be a set containing the $|Y|$-dimensional standard basis vectors, $U := \mathrm{conv}(E)$, and $D := \{\delta \in \mathbf{R}^{|Y|} \mid \delta^T \mathbf{1} = 0, \|\delta\|_\infty \leq 1\}$. For any vector $\delta \in D$, and*

[5] A vector u_i can be obtained by minimising $\min_{u \in U} \|x_i - Wu\|_2^2$. The non-uniqueness of the solution comes from Carathéodory's theorem—see, for example, [Ber60].

sequence $\{u_i\}_{i=1}^{|Y|}$ *of points from* U*, there exists at least one sequence* $\{e_i\}_{i=1}^{|Y|}$ *of points from* E *such that*

$$(\delta + z - z') \in D, \tag{4.8}$$

where $z := \sum_{i=1}^{|Y|} u_i$ *and* $z' := \sum_{i=1}^{|Y|} e_i$*. That is,* $\mathbf{1}^T(\delta + z - z') = 0$ *and* $\|\delta + z - z'\|_\infty \le 1$*.*

Proof. To start, let $V = |Y|$ and note we always have $\mathbf{1}^T(z + \delta) = V$ since z is the sum of V elements from U, $u^T\mathbf{1} = 1$ and $\delta^T\mathbf{1} = 0$ for all $u \in U$, $\delta \in D$. Further, $(z + \delta) \succeq -\mathbf{1}$ since $\delta \succeq -\mathbf{1}$ and $z \succeq 0$. Now, let $r := (z + \delta)$ and define

$$a = -[-r]^+, \quad b = \lfloor r - a \rfloor, \quad c = r - a - b,$$

where the floor in b is taken element-wise. That is, $a \in [-1, 0]^V$, $b \in \{0, 1, \dots, V\}^V$, $c \in [0, 1)^V$. For example, if $r = [2.2, -0.2]^T$ then $a = [0, -0.2]^T$, $b = [2, 0]^T$ and $c = [0.2, 0]^T$. Observe,

$$\mathbf{1}^T r = \mathbf{1}^T(a + b + c) = V,$$

and since b is integer valued $\mathbf{1}^T b \in \mathbf{Z}_+$, which implies $\mathbf{1}^T(a + c) \in \mathbf{Z}_+$. Next, let $\mathbf{1}^T b = V - \mathbf{1}^T(a + c) := V'$, and observe b can be written as the sum of V' elements from E, *i.e.*

$$b = \sum_{i=1}^{V'} e_i.$$

Next, since $-\mathbf{1} \preceq a + c \prec \mathbf{1}$ and $\mathbf{1}^T(a + c) = V - V' := V''$, there must exist *at least* V'' elements in vector $(a + c)$ that are nonnegative. If we select V'' elements from E that match the non-negative components of vector $(a + c)$ we can construct a subsequence $\{e_i\}_{i=1}^{V''}$ such that

$$-\mathbf{1} \preceq (a + c) - \sum_{i=1}^{V''} e_i \prec \mathbf{1}.$$

Finally, letting $z' = \sum_{i=1}^{V'} e_i + \sum_{i=1}^{V''} e_i$ yields the result. ■

Lemma 4.2 says that for any vector $\delta_1 \in D$ and sequence $\{u_i^{(1)}\}_{i=1}^{|Y|}$ of points from U, there exists at least one sequence $\{e_i^{(1)}\}_{i=1}^{|Y|}$ of points from E such that the sum of the elements in the sequences and an "offset" δ_1 lies in D. That is, if we let z_1 and z_1' be, respectively, the sum of the elements in the sequences from U and E we have that $(\delta_1 + z_1 - z_1') = \delta_2 \in D$. Similarly, for another sequence $\{u_i^{(2)}\}_{i=1}^{|Y|}$ of points from U, we can construct another sequence $\{e_i^{(2)}\}_{i=1}^{|Y|}$ of points from E such that the sum of the elements in the sequences and an "offset" δ_2 lies again in D. That is, $(z_2 - z_2' + (\delta_1 + z_1 - z_1')) \in D$

where z_2 and z_2' are, respectively, the sum of the elements in sequences $\{u_i^{(2)}\}_{i=1}^{|Y|}$ and $\{e_i^{(2)}\}_{i=1}^{|Y|}$. Hence, for any sequence $\{u^{(\tau)}\}_{i=1}^{|Y|}$, $\tau = 1, 2, \dots$ we can construct a sequence $\{e_i^{(\tau)}\}_{i=1}^{|Y|}$ such that their sum plus an offset δ_τ lies in D. That is,

$$\begin{aligned} &\big(((\cdots((\delta_1 + z_1 - z_1') + z_2 - z_2') + \cdots + z_{K-1} - z_{K-1}') + z_K - z_K')\big) \\ &= \left(\delta_1 + \sum_{\tau=1}^{K} z_\tau - z_\tau'\right) \in D. \end{aligned} \tag{4.9}$$

where z_τ and z_τ' are the sum of the elements in sequences $\{u^{(\tau)}\}_{i=1}^{|Y|}$ and $\{e^{(\tau)}\}_{i=1}^{|Y|}$, $\tau = 1, 2, \dots, K$. It follows that $\| \sum_{i=1}^{k} u_i - e_i \|_\infty \leq 1$ for $k \in \tau|Y|$, $\tau \in \mathbf{Z}_+$.

One can note from (4.9) the inherited "renewal" property of these types of sequences, where the construction of a (sub)sequence can be carried out independently of the previous (sub)sequences since the resulting offset always lies in D. Furthermore, since all we care about is the sum of the subsequences, the elements in $\{e^{(\tau)}\}_{i=1}^{|Y|}$ can be reordered in order to match or model a meta-action in a system. It is important to highlight that Lemma 4.2 does not say how to construct such sequences, it only establishes the existence of sequences with such properties. However, update

$$e_i^{(\tau)} \in \arg\min_{e \in E} \left\| \left(\delta_\tau + z_\tau - \sum_{\kappa=1}^{i-1} e_\kappa^{(\tau)}\right) - e \right\|_\infty \qquad i = 1, \dots, |Y|, \tag{4.10}$$

with $\delta_\tau = \delta_1 + \sum_{\tau=1}^{K-1}(z_\tau - z_\tau')$, $K = 1, 2, \dots$ yields a sequence $\{e_i^{(\tau)}\}_{i=1}^{|Y|}$ that satisfies the properties of Lemma 4.2.[6] We show this is the case with the following lemma for a single iteration, but the argument can be applied recursively.

LEMMA 4.3. *Consider the setup of Lemma 4.2 and select* $e_i \in \arg\min_{e \in E} \|(\delta + z - \sum_{\kappa=1}^{i-1} e_\kappa) - e\|_\infty$, $i = 1, \dots, |Y|$. *Then,* $-\mathbf{1} \preceq \delta + z - z' \preceq \mathbf{1}$, *where* $z' = \sum_{i=1}^{|Y|} e_i$.

PROOF. First of all, recall from the proof of Lemma 4.2 that $\mathbf{1}^T(\delta + z - z') = 0$, $\delta + z \succeq -\mathbf{1}$ and that $V := \mathbf{1}^T(z + \delta)$ where $V := |Y|$. Next, define $r_i := \delta + z - \sum_{\kappa=1}^{i-1} e_\kappa$, $i = 1, 2, \dots$ and note that update $e_i \in \arg\min_{e \in E} \|r_i - e\|_\infty$ decreases the largest component of vector r_i, *i.e.* in each iteration a component of vector r_i decreases by 1, and therefore $\mathbf{1}^T r_i = V - i + 1$ with $i = 1, \dots, |Y| + 1$.

For the lower bound observe that if $r_{i+1}(j) < -1$ for some $j = 1, \dots, |Y|$ we must have that $r_i \prec 0$ since the update $e_i \in \arg\min_{e \in E} \|r_i - e\|_\infty$ selects to decrease the largest component of vector r_i. However, since $\mathbf{1}^T r_i \geq 0$ for all $i = 1, \dots, |Y| + 1$ we have that vector $r_{|Y|}$ has at least one component that is nonnegative. Therefore, $r_{|Y|+1} \succeq -\mathbf{1}$ and $\delta + z - z' \succeq -\mathbf{1}$. For the upper bound define $a_i = -[-r_i]^+$, $b_i = \lfloor r_i - a_i \rfloor$, $c_i = r_i - a_i - b_i$,

[6] The update is in spirit very similar to update (4.16) in Theorem 4.2.

$i = 1, \dots, |Y|+1$ and note that $-\mathbf{1} \preceq a_i \preceq 0$ and $0 \preceq c_i \prec \mathbf{1}$ for all $i = 1, \dots, |Y|+1$, and that $\mathbf{1}^T b_i$ decreases by 1 in each iteration if $\mathbf{1}^T b_i \geq 1$. Hence, $b_{|Y|+1} = 0$ and therefore $-\mathbf{1} \preceq r_{|Y|+1} = a_{|Y|+1} + c_{|Y|+1} = \delta + z - z' \preceq \mathbf{1}$ and we are done. ■

An important observation from this lemma is that it only allows the construction of a sequence $\{e_i^{(\tau)}\}_{i=1}^{|Y|}$ when a sequence $\{u_i^{(\tau)}\}_{i=1}^{|Y|}$ is known. Hence, a sequence $\{e_i^{(\tau)}\}_{i=1}^{|Y|}$ cannot be constructed in an online manner. Nonetheless, our interest now is not to give a specific algorithm that satisfies the properties of Lemma 4.2, but to develop the notion of a block of discrete actions, that will later allow us to construct sequences of actions in a flexible manner. Consider the following example.

Example 4.1. *We show using a simulation that by using update (4.10) we can generate a sequence $\{e_k\}$ that keeps $\| \sum_{i=1}^k u_i - e_i \|_2$ uniformly bounded. We let $|Y| \in \{10, 100, 1000, 5000\}$ and generate a sequence of points from U uniformly at random. Update (4.10) is used to generate subsequences of points from E of length $|Y|$. The results from the simulations are shown in Figure 4.3.1. Observe from the figure that as we increase the number of points in Y, it is easier to see that $\| \sum_{i=1}^k u_i - e_i \|_2$ is attracted to a smaller ball when $k = \tau |Y|$, $\tau \in \mathbf{N}$. This behaviour is a consequence of Lemma 4.2 and the concatenation of (sub)sequences as shown in (4.9). We will consider this behaviour in more detail shortly.*

We are now in position to present the following theorem, which establishes a bound on $\| \sum_{i=1}^k u_i - e_i \|_2$ for a sequence of (sub)sequences that comply with the conditions of Lemma 4.2.

Theorem 4.1. *Consider the setup of Lemma 4.2 with $\delta = 0$, and suppose we have sequences $\{u_k\}$ and $\{e_k\}$ from U and E such that*

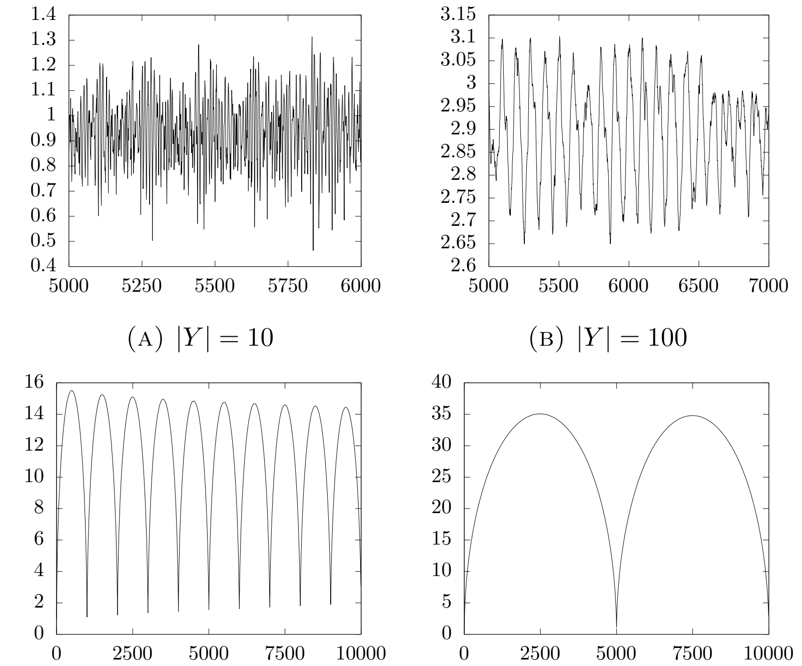
$$\left(\sum_{i=1}^k u_i - e_i \right) \in D \qquad k \in \tau |Y|, \tau = 1, 2, \dots,$$

Then,

- $\| \sum_{i=1}^k u_i - e_i \|_2 \leq \sqrt{|Y|}$ *when* $k = \tau |Y|$, $\tau \in \mathbf{N}$
- $\| \sum_{i=1}^k u_i - e_i \|_2 \leq \sqrt{|Y|} + 2|Y|$ *for all* $k = 1, 2, \dots$

Proof. Recall that z_τ and z'_τ are, respectively, the sum of the elements in subsequences $\{u^{(\tau)}\}_{i=1}^{|Y|}$ and $\{e^{(\tau)}\}_{i=1}^{|Y|}$. Since $(\sum_{\tau=1}^K z_\tau - z'_\tau) \in D$ we have that

$$\left\| \sum_{\tau=1}^K z_\tau - z'_\tau \right\|_\infty = \left\| \sum_{\tau=1}^K \sum_{i=(\tau-1)|Y|+1}^{(\tau-1)|Y|+|Y|} u_i - e_i \right\|_\infty \leq 1 \tag{4.11}$$



(A) $|Y| = 10$ (B) $|Y| = 100$

(C) $|Y| = 1000$ (D) $|Y| = 5000$

FIGURE 4.3.1. Illustrating bound $\|\sum_{i=1}^{k} u_i - e_i\|_2$ of Example 4.1.

and therefore

$$\left\| \sum_{\tau=1}^{K} \sum_{i=(\tau-1)|Y|+1}^{(\tau-1)|Y|+|Y|} u_i - e_i \right\|_2 \leq \sqrt{|Y|} \qquad K = 1, 2, \ldots \tag{4.12}$$

Next, observe that

$$\begin{aligned}
\left\| \sum_{\tau=1}^{K} \sum_{i=(\tau-1)|Y|+1}^{(\tau-1)|Y|+|Y|} u_i - e_i \right\|_2 &\leq \left\| \sum_{\tau=1}^{K-1} z_\tau - z'_\tau \right\|_2 + \left\| \sum_{i=(K-1)+1}^{K|Y|} u_i - e_i \right\|_2 \\
&\leq \sqrt{|Y|} + \left\| \sum_{i=(K-1)+1}^{K|Y|} u_i - e_i \right\|_2 \\
&\leq \sqrt{|Y|} + \sum_{i=(K-1)+1}^{K|Y|} \|u_i - e_i\|_2 \\
&\leq \sqrt{|Y|} + |Y| \max_{u \in U, e \in E} \|u - e\|_2 \\
&\leq \sqrt{|Y|} + 2|Y|
\end{aligned} \tag{4.13}$$

where the last equation follows since

$$\max_{u \in U, e \in E} \|u - e\|_2 \leq \max_{u \in U, e \in E} (\|u\|_2 + \|e\|_2) \leq 2.$$

To conclude that the bound holds for any $k \in \mathbf{N}$, it is sufficient to note in (4.13) that

$$\|u_1 - e_1\|_2 \leq \sum_{i=1}^{2} \|u_i - e_i\|_2 \leq \cdots \leq \sum_{i=1}^{|Y|-1} \|u_i - e_i\|_2 \leq \sum_{i=1}^{|Y|} \|u_i - e_i\|_2$$

and we are done. ■

Important observations from Theorem 4.1 are that the bound depends on the number of points in set Y, and that the difference between sequences is periodically attracted to a ball around $\sqrt{|Y|}$ when $k = \tau|Y|$, $\tau \in \mathbf{N}$.

The selection of a sequence $\{e_i\}_{i=1}^{|Y|}$ for each sequence $\{u_i\}_{i=1}^{|Y|}$ in Lemma 4.2 can be generalised to selecting a sequence $\{e_i\}_{i=1}^{T|Y|}$ for every sequence $\{u_i\}_{i=1}^{T|Y|}$ with $T \in \mathbf{N}$. To see this is always possible, it is sufficient to note that if we know a full sequence $\{u_i\}_{i=1}^{T|Y|}$, the problem can be split in a sequence of T subproblems. We have the following corollary.

Corollary 4.2. *Consider the setup of Lemma 4.2 with $\delta = 0$, and suppose we have sequences $\{u_k\}$ and $\{e_k\}$ from U and E such that*

$$\left(\sum_{i=1}^{k} u_i - e_i\right) \in D \qquad k \in \tau|Y|,\ T \in \mathbf{N},\ \tau = T, 2T, 3T, \ldots$$

Then,

$$\left\|\sum_{i=1}^{k} u_i - e_i\right\|_2 \leq \sqrt{|Y|} + 2T|Y|. \tag{4.14}$$

The most important point from this section is that we can always construct a sequence of discrete actions that breaks free from the past every $T|Y|$ iterations, and that the bound on that sequence depends on the number of points in set $|Y|$. These two points are key when designing sequences of actions that keep the difference $\|\sum_{i=1}^{k} u_i - e_i\|_2$ bounded.

The construction of sequences of discrete actions using blocks (groups of actions) will be very useful in some cases, but in others, might be over complicated, especially when the number of actions is large. In the next section, we present some algorithms that do not depend on blocks, and allow sequences of discrete actions to be constructed in an online manner.

4.3.2. Online Sequences of Discrete Actions. We start by presenting the following theorem, which shows how a sequence can be constructed using a "greedy" update.

THEOREM 4.2. *Let $\{x_k\}$ be a sequence of points from $X \subseteq \mathrm{conv}(Y)$ where Y is a finite collection of points from $\mathbf{R}^n$. Select*

$$u_k \in \arg\min_{u \in U} \|Wu - x_k\|_2^2, \tag{4.15}$$

$$e_k \in \arg\min_{e \in E} \|s_{k-1} + u_k - e\|_\infty, \tag{4.16}$$

$$y_k = We_k, \tag{4.17}$$

where $s_k = \sum_{i=1}^{k}(u_i - e_i)$. Then, we have that $-\mathbf{1} \preceq s_k \preceq (|Y|-1)\mathbf{1}$, and

$$\left\| \sum_{i=1}^{k} x_i - y_i \right\|_2 \leq C \tag{4.18}$$

where $C := \sqrt{|Y|}(|Y|-1)\|W\|_2$.

PROOF. We begin by noting that since $\mathbf{1}^T u_k = 1 = \mathbf{1}^T e_k$ then $\mathbf{1}^T s_k = 0$ for all $k = 1, 2, \ldots$. Also note that since $u_k \in U$ all elements of u_k are non-negative and at least one element must be non-zero since $\mathbf{1}^T u_k = 1$.

We now proceed by induction to show that there always exists a choice of e_{k+1} such that $s_k \succeq -\mathbf{1}$, $k = 1, 2, \ldots$. When $k = 1$ let element $u_1(j)$ be positive (as already noted, at least one such element exists). Selecting $e_1 = v_j$ then it follows that $-1 < u_1(j) - e_1(j) \leq 0$ and so $-1 \prec u_1 - e_1 \prec 1$. That is, $s_1 \succeq -\mathbf{1}$. Suppose now that $s_k \succ -\mathbf{1}$. We need to show that $s_{k+1} \succeq -\mathbf{1}$. Now $s_{k+1} = s_k + u_{k+1} - e_{k+1}$. Since $s_k \succeq -\mathbf{1}$, $s_k(j) \geq -1\ \forall j = 1, \ldots, |Y|$. Also, $\mathbf{1}^T s_k = 0$, so either all elements are 0 or at least one element is positive. If they are all zero then we are done (we are back to the $k = 1$ case). Otherwise, since all elements of u_{k+1} are non-negative then at least one element of $s_k + u_{k+1}$ is positive. Let element $s_k(j) + u_{k+1}(j)$ be the largest positive element of $s_k + u_{k+1}$. Selecting $e_{k+1} = v_j$ then it follows that $s_k(j) + u_{k+1}(j) - e_{k+1}(j) \geq -1$. That is, $s_{k+1} \succeq -\mathbf{1}$.

We now show that s_k is upper bounded. Recall e_{k+1} can always be selected such that $s_k \succeq -\mathbf{1}$, and also $\mathbf{1}^T s_k = 0$. Since $\mathbf{1}^T s_k = 0$ either s_k is zero or at least one element is positive. Since $\boldsymbol{s}_k \succeq -\mathbf{1}$ and at most $|Y| - 1$ elements are negative, then the sum over the negative elements is lower bounded by $-(|Y| - 1)$. Since $\mathbf{1}^T s_k = 0$ it follows that the sum over the positive elements must be upper bounded by $|Y| - 1$. Hence, $\|s_k\|_\infty \leq (|Y| - 1)$.

Finally, observing that

$$\begin{aligned}\left\| \sum_{i=1}^{k} x_i - y_i \right\|_2 \leq \|W\|_2 \left\| \sum_{i=1}^{k} u_i - e_i \right\|_2 &\leq \sqrt{|Y|}\|W\|_2 \left\| \sum_{i=1}^{k} u_i - e_i \right\|_\infty \\ &\leq \sqrt{|Y|}(|Y|-1)\|W\|_2\end{aligned}$$

we are done. ∎

Theorem 4.2 guarantees that by using updates (4.15)-(4.17) the difference (4.7) is uniformly bounded. Observe that update (4.16) actually corresponds to selecting a vector $e \in E$ that decreases the largest component of vector s_k, and does not provide any degree of flexibility as to how to select other actions in Y. Regarding the complexity of the update, note that (4.17) consists of selecting a column of matrix W and so it is computationally inexpensive. However, updates (4.15) and (4.16) involve solving a convex programme and using exhaustive search, so some care is required to ensure that the updates can be efficiently performed in each time slot. Hence, if Y and X can be decomposed in orthogonal subsets will play an important role in terms of scalability.

Next we present the following corollary to Theorem 4.2, which allows us to select an action y_k for a consecutive sequence of time slots.

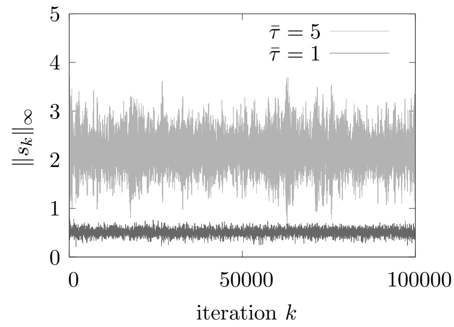
Corollary 4.3. *Consider the setup of Theorem 4.2 where update (4.15) is performed in each iteration. Updates (4.16)-(4.17) are performed in time slots* $\{\tau_1, \tau_2, \dots\} := \mathcal{T} \subseteq \mathbf{N}$*; otherwise,* e_k *and* y_k *are selected equal to* e_{k-1} *and* y_{k-1}*. Then, we have that* $-\bar{\tau}\mathbf{1} \preceq s_k \preceq \bar{\tau}(|Y|-1)\mathbf{1}$*, for all* k *where* $\bar{\tau} = \max_{j\in\{1,2,\dots\}}\{\tau_{j+1} - \tau_j\}$ *and*

$$\left\| \sum_{i=1}^{k} x_i - y_i \right\|_2 \leq \bar{\tau} C. \tag{4.19}$$

Proof. Since s_k has at least one component that is non-negative, and update (4.16) selects the largest component of vector s_k when $k \in \mathcal{T}$, we have that a component of vector s_k can decrease at most by $\bar{\tau}$ in an interval $\{\tau_j - \tau_{j+1}\}$ for all $j = 1, 2, \dots$. Hence, $s_k \succeq -\bar{\tau}\mathbf{1}$ for all k. Next, since $s_k^T \mathbf{1} = 0$ for all k and the sum over the negative components is at most $-\bar{\tau}(|Y|-1)$, we have that $s_k \preceq \bar{\tau}(|Y|-1)\mathbf{1}$. The rest of the proof follows as in Theorem 4.2. ■

Corollary 4.3 says that the difference s_k in Theorem 4.2 will be bounded when the difference between the times when updates (4.15)-(4.17) are performed is bounded. This corollary will be useful when taking an action implies that it has to be held for a number of iterations. For example, if we want to model the meta-action of transmitting a packet and a discrete action represents transmitting a bit, this corollary ensures that a sequence of bits can be transmitted until the whole packet has been sent. The condition that $\bar{\tau}$ must be finite corresponds in this case to requiring that packets have finite length. Figure 4.3.2 shows an example of Corollary 4.3 with $\bar{\tau} \in \{1, 5\}$. In the example Y is a set of 10 points generated uniformly at random from $\mathbf{R}^5$, $X = \text{conv}(Y)$ and $\{x_k\}$ a random[7] sequence from X. Observe from the figure that $\|s_k\|_\infty$ is bounded for all values $\bar{\tau}$.

[7]For simplicity, we generate a sequence $\{u_k\}$ uniformly at random. Recall $x_k = W u_k$.



(A)

FIGURE 4.3.2. Illustrating Corollary 4.3 with $\bar{\tau} \in \{1, 5\}$. Action set Y consists of 10 points generated uniformly at random from $\mathbf{R}^5$.

We are interested in obtaining general sequences of actions that are flexible and can be easily adapted to the requirements of different problems. It is difficult to give a specific algorithm without specifying a problem, nevertheless, we can establish the conditions that a generic algorithm should check when selecting a sequence of actions. As shown in the previous section, for finite $|Y|$ it is possible to construct a sequence of actions that breaks free from the past for a subsequence that is sufficiently large. The exact same concept can be applied in this case, but now we must ensure that $\|s_k\|_\infty$ is bounded for all k. This motivates the following theorem.

THEOREM 4.3. *Let $\{x_k\}$ be a sequence of points from $X \subseteq \text{conv}(Y)$ where Y is a finite collection of points from $\mathbf{R}^n$. For any sequence $\{y_k\}$ of points from Y we have that*

$$\left\| \sum_{i=1}^{k} x_i - y_i \right\|_2 \leq \gamma_k C \tag{4.20}$$

where $\gamma_k = -\min_{j \in \{1,\ldots,|Y|\}} s_k(j)$, $s_k = \sum_{i=1}^{k} u_i - e_i$, $u_k \in \arg\min_{u \in U} \|Wu - x_k\|_2$, and $C = \sqrt{|Y|}(|Y| - 1)\|W\|_2$.

PROOF. Recall that since $\mathbf{1}^T u_k = 1 = \mathbf{1}^T e_k$ then $\mathbf{1}^T s_k = 0$ for all $k = 1, 2, \ldots$, and therefore s_k is either 0 or at least one of its components is strictly positive. Next, observe that since $\gamma_k = -\min_{j \in \{1,\ldots,|Y|\}} s_k(j)$ we have that $\max_{j \in \{1,\ldots,|Y|\}} s_k(j) \leq \gamma_k(|Y| - 1)$, which corresponds to the case where $|Y| - 1$ components of vector of vector s_k are equal to γ_k. The rest of the proof continues as in the proof of Theorem 4.2. ■

Theorem 4.3 says that when we can construct a sequence of actions $\{y_k\}$, such that γ_k is bounded then the difference $\| \sum_{i=1}^{k} x_i - y_i \|_2$ will be bounded. In order to make sure that γ_k is bounded we need to compute vector s_k at each iteration, which involves obtaining a vector u_k (solving a quadratic programme) and a vector e_k. However, now e_k does not need to be obtained as in (4.16) as long as it is done with some "care". Namely, by not selecting actions that decrease lower bound γ_k "excessively". For example, selecting a vector e_k that decreases a positive component of vector s_k will be enough. The implications of this are important in terms of scalability, because when action set Y is large, we do not need to do exhaustive search over all the elements to select a vector from E.

CHAPTER 5

Applications

5.1. Traffic Signal Control

Traffic signal control is an active area of research with a high social impact in urban areas. In short, the problem consists of designing a policy for the traffic control system in a city/area that reduces the network congestion while considering all the actors of the transportation system (pedestrians, vehicles, *etc.*). Many cities currently use adaptive traffic signal control systems such as SCATS [Low90], SCOOT [HRBW81], OPAC [Gar83], RHODES [MH01], *etc.* that adapt their behaviour depending on the network conditions and a set of predefined parameters. Although these systems have been shown to obtain a better performance than traditional systems [Pap80] (which adapt their behaviour depending on the time of the day), their performance is still poorly understood since it depends greatly on the network structure and parameters used.

In recent there has been increasing attention to model urban traffic networks as a network of interconnected queues, and to view traffic signal control decisions as akin to packet scheduling decisions in communication networks [WUW+12]. This change of paradigm has been mostly motivated by the fact that (i) it is then possible to characterise the capacity of a network (*i.e.* the maximum flow of vehicles that a network can handle), and (ii) it is possible to use existing scheduling algorithms from communications networks for traffic signal control. Amongst those algorithms stands out max-weight scheduling [TE92] (also known in the literature as backpressure routing) because it can maximise the network throughput without previous statistical knowledge of the underlying "randomness" in the network. Further, max-weight scheduling makes decisions based only on the current network state and does not require any set of predefined parameters.

Besides the strong similarities between communication and urban traffic networks, there are some well-known issues that arise in urban traffic networks that are not present in communications networks. For instance, queues (roads) in traffic networks have limited length, traffic lights can give the right of way but cannot make routing decisions, and traffic network controllers might not have perfect information about how many vehicles are in each traffic light waiting to be served. The aforementioned issues are well known by the community and have been addressed in previous work (see for example [GQF+15]

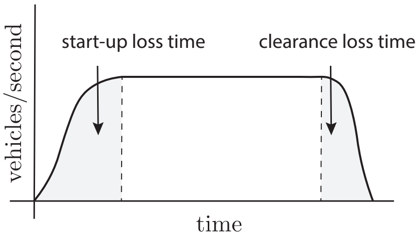


FIGURE 5.1.1. Schematic illustration of the number of vehicles per second that can get through a intersection during a phase.

and [GFDLFW13]). However, there are still some issues that remain open. For example, max-weight approaches do not integrate the notion of traffic light cycle in intersections, which is fundamental in many operational traffic signal control strategies. As a result, it is poorly understood how max-weight approaches compare to classic traffic signal control approaches.

Another important issue that has not been considered in previous work is that the order in which traffic control actions are taken affects the performance of the network. In brief, suppose the time in the system is divided into time slots of equal duration, and that at each time slot the network controller decides which phase to activate in an intersection. Namely, which traffic lights in an intersection should be red or green. During a phase, a certain number of vehicles go though the intersections, however, the rate of vehicles is not constant. When a traffic light changes from red to green, vehicles need a certain time to reach the desired speed (start-up loss time); and when the traffic light changes back to red, some time is required to clear the intersection (clearance loss time). Figure 5.1.1 schematically shows how the rate of vehicles that go through an intersection depends on time. Observe from the figure that the start-up and clearance loss times are in fact an overhead that reduces the *average* rate of vehicles that can go through the intersection. Note, however, that when the same phase is selected consecutively, the start-up and clearance time overheads remain constant since there is no need of stopping/resuming the flow of vehicles in the intersection. This observation motivates the design of traffic signal control policies where phases are selected consecutively, or equivalently, that reduce the frequency in which traffic lights change state. The design of such policies is however not possible with the current max-weight approaches in the literature since these strongly link the stability of the system to the use of a specific policy.

In this section, we present a new approach to traffic signal control based on convex optimisation. Our approach is simpler than other approaches in the literature, and allows us to separate the stability of the system from the construction of a traffic control policy.

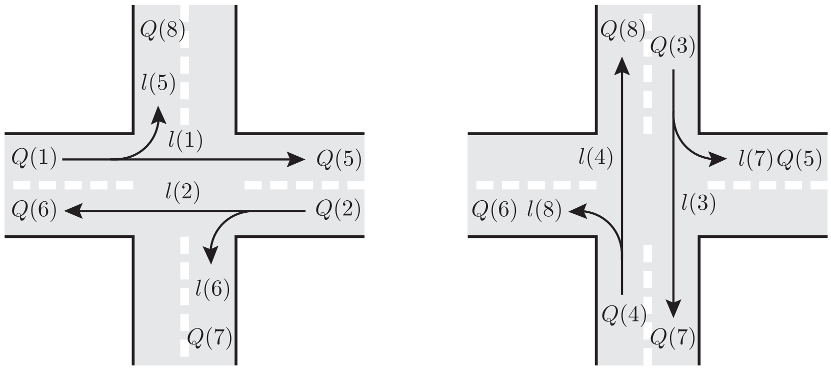
As a result, we can design a range of policies that maximise the network throughput, and that at the same time can capture urban traffic networks characteristics more accurately. With an example, we show how a traffic signal control policy affects the network capacity region and the delay distribution of the vehicles.

5.1.1. Preliminaries.

Model. We model a traffic network as a directed graph of n nodes and l links, where a node represents a queue/road[1] and a link is a connection between two roads in, for example, an intersection. The time in the system is slotted and each time slot $k = 1, 2, \dots$ has equal duration. The dynamics of the queues in our system are given by update $Q_{k+1} = [Q_k + \delta_k]^+,\ k = 1, 2, \dots$ where $Q_k \in \mathbf{Z}^n_+$ and $\delta_k \in \mathbf{Z}^n$ is a vector that contains the queues net increments. That is, $\delta_k(j) \in \mathbf{Z},\ j = 1, \dots, n$ contains the difference of vehicles that get put in and taken out of $Q_k(j)$ from time slot k to time slot $k + 1$. We will represent the connection between queues using incidence matrix $A \in \{-1, 0, 1\}^{n \times l}$ where -1 indicates that a link is leaving a node; 1 that a link is entering a node; and 0 that a node and a link are not connected. For example, a -1 in the j'th element of the i'th row of matrix A indicates that link j is leaving node i. An intersection in a network is represented as a collection of nodes and links, and a *phase* in an intersection is a set of non-conflicting links (which connect the roads in the intersection) that can be simultaneously active. Importantly, phases in different intersections are always sets of non-conflicting links, *i.e.* a phase in an intersection can be activated independently of the phase selected in another intersection. Figure 5.1.2 shows an example of a standard intersection of 8 nodes (roads), 8 links (connections between roads) and two phases (set of non-conflicting links that can be simultaneously active).

The operation of the network is as follows. Vehicles can arrive and leave the network in every queue/road (*e.g.* arriving or leaving a parking space), and at each time slot $k = 1, 2, \dots$ the network controller selects which links in the network can be active. More precisely, the network controller selects a vector $y \in Y \subseteq \{0, 1\}^l$ where a 1 in the i'th element of the vector indicates that the i'th link is active and 0 that it is not. For example, phase 1 and 2 in the intersection shown in Figure 5.1.2 correspond, respectively, to selecting actions $[1, 1, 0, 0, 1, 1, 0, 0]^T$ and $[0, 0, 1, 1, 0, 0, 1, 1]^T$. Hence, at each time slot we have the update $Q_{k+1} = [Q_k + Ay_k + B_k]^+$, where Ay_k captures how vehicles move from one queue/road to another, and $B_k \in \mathbf{Z}^n$ is the net increment of vehicles that arrive/leave

[1]As presented in [GFDLFW13], a road can contain multiple queues, however, we will use only one queue per road for simplicity of exposition since the extension is straightforward.



(A) Phase 1 (B) Phase 2

FIGURE 5.1.2. Illustrating two typical phases in an intersection. Phase 1 gives the right of way to $Q(1)$ and $Q(2)$; and phase 2 gives right of way to $Q(3)$ and $Q(4)$.

the system. For simplicity we will assume that in each time slot exactly one vehicle moves from one queue to another.

Stability and Capacity Region. Before considering the traffic signal policy design, we need to first define the concepts of stability and capacity region, which will allow us to determine whether a traffic signal control policy is optimal or not. Stability can be informally regarded as all the "work" that gets put into a queue gets eventually serviced, which, in the context of urban traffic networks, is equivalent to saying that a vehicle will reach its destination in finite time. There are different definitions of stability in the literature, however, by stability we will mean *strong stability*[2] [Nee10a], which we define next.

DEFINITION 5.1 (System Stability). *We say that a queuing system is stable if*

$$\lim_{k\to\infty} \mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k} Q_i\right) \prec \infty \tag{5.1}$$

where $Q_k \in \mathbf{Z}_+^n$.

The network capacity region relates the network traffic load with the existence of a policy that can stabilise the system.

DEFINITION 5.2 (Network Capacity Region). *Set of mean vehicle arrival/departure rates in the system such that there exists a traffic signal control or scheduling policy that can keep a queuing system stable.*

Figure 5.1.3 schematically shows the capacity region of the intersection depicted in Figure 5.1.2 for two policies. Observe from the figure that we have added a 'phase 0' to

[2]Under some boundedness assumption strong stability implies also the other forms of stability [Nee10a].

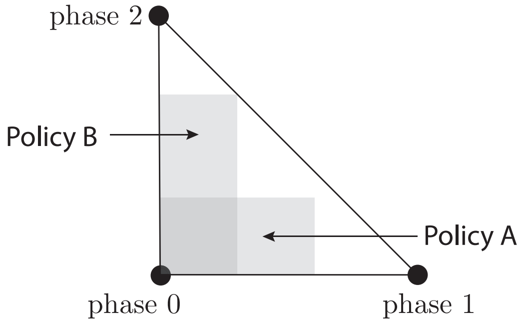


FIGURE 5.1.3. Schematic illustration of the capacity region of two policies. Policy A uses phases 0, 1 and 2, respectively, 10 %, 60 % and 30 % of the time, whereas Policy B uses phases 1 and 2, respectively, 30 % and 70 % of the time.

the intersection to capture the fact that neither phase 1 nor phase 2 are active, *i.e.* there is no flow of vehicles in the intersection[3]. A policy in an intersection selects which fraction of time each phase is selected. In this example Policy A uses phases/actions 0, 1 and 2, respectively, 10 %, 60 % and 30 % of the time, whereas Policy B uses actions/phases 1 and 2, respectively, 30 % and 70 % of the time. Each policy can stabilise the system when the mean vehicle arrival/departure rate in the system is within their respective "capacity regions" (shaded areas). For example, if an activated phase corresponds to serving one vehicle per time slot, Policy B stabilises the system when the *mean* vehicle arrival rate to $Q(1)$ and $Q(2)$ is less than or equal to 0.3 vehicles per time slot, and less than or equal to 0.7 vehicles for $Q(3)$ and $Q(4)$.

In the next section, we present our convex optimisation approach with stochastic vehicles arrivals and discrete actions.

5.1.2. Convex Optimisation Approach. Consider the model presented in Section 5.1.1 and the following convex optimisation problem $\mathcal{P}$:

$$\begin{aligned} \underset{x\in X}{\text{minimise}} \quad & 1 \\ \text{subject to} \quad & Ax \preceq b \end{aligned}$$

where $A \in \mathbf{R}^{n\times l}$, $b \in \mathbf{R}^n$, X a convex subset from $\text{conv}(Y)$, and $Y := \{y_1, \dots, y_{|Y|}\} \subseteq \{0,1\}^l$ is a set that contains which links in the network can be simultaneously selected. Recall that as defined in Section 5.1.1 matrix A represents the interconnection between nodes (roads) and links, and b is the mean vehicle arrival/departure rate of the system in each of the roads. We will always assume that $X_0 := \{x \in X \mid Ax \preceq b\}$ has non-empty

[3]In Section 5.1.3 we will show that phase 0 can be used to capture the time loss when changing between phases.

relative interior[4], *i.e.* the Slater conditions is satisfied (Assumption 3.2) and strong duality holds.

Optimisation problem $\mathcal{P}$ can be solved using standard convex optimisation methods if b were known, however, this is not typically the case in real networks. We aim to design a traffic signal control policy that can stabilise the system and does not require to know b as long as X_0 has a non-empty relative interior. This will be possible by using the dual formulation of problem $\mathcal{P}$ shown in the framework in Section 3.3.

Consider the following maximisation problem $\mathcal{D}$:

$$\underset{\lambda \succeq 0}{\text{maximise}} \quad h(\lambda) \tag{5.2}$$

where $h(\lambda)$ is the Lagrange dual function defined as

$$h(\lambda) := \min_{x \in X} L(x, \lambda) \tag{5.3}$$

$$= \min_{x \in X} \{1 + \lambda^T (Ax - b)\}, \tag{5.4}$$

and $\lambda \in \mathbf{R}^n_+$ is a vector of Lagrange multipliers. Since $h(\lambda)$ is concave (see [BV04, Chapter 5]) we can use the subgradient method to solve the dual problem [Ber99], which consists of the following update:

$$\lambda_{k+1} = [\lambda_k + \alpha_k \partial h(\lambda_k)]^+ = [\lambda_k + \alpha_k (Ax_k - b)]^+, \tag{5.5}$$

where $\lambda_1 \in \mathbf{R}^n_+$, $\partial h(\lambda_k)$ is the subgradient of the dual function at point λ_k, and $\alpha > 0$ is a step size. In this case, since we have a feasibility problem (no objective function) it will be sufficient to fix $\alpha = 1$ and work with simpler update

$$\lambda_{k+1} = [\lambda_k + Ax_k - b]^+. \tag{5.6}$$

A key characteristic of the subgradient method is that the value x_k used in (5.6) can be obtained independently of b. In particular, by solving the following linear (so convex) optimisation problem

$$\begin{aligned} x_k &\in \arg\min_{x \in X} L(x, \lambda_k) \\ &= \arg\min_{x \in X} \{1 + \lambda_k^T (Ax - b)\} \\ &= \arg\min_{x \in X} \{\lambda_k^T Ax\}. \end{aligned} \tag{5.7}$$

[4]This assumption is known in convex optimisation as the Slater condition (see [Ber99, Proposition 3.3.9]), and in max-weight it is equivalent to requiring that the mean arrival rate is in the interior of the network capacity region.

We now proceed to relax the use of b in update (5.6). Observe that the Lagrange multiplier update (5.6) resembles a queue update, however, with the difference that the input/output of the queue is real valued instead of discrete. Informally, instead of having vehicles moving from one queue to another or arriving/leaving the system at each time slot, we have that fractions of vehicles move from one queue to another and arrive/leave the system. We can consider a stochastic version of the subgradient method for the dual problem and change b in (5.6) for a random variable $B_k \in \mathbf{Z}^n$ with mean $\mathbf{E}(B_k) = b$, *i.e.* we have update

$$\lambda_{k+1} = [\lambda_k + Ax_k - B_k]^+. \tag{5.8}$$

As a result, now vehicles (and not fractions of vehicles) arrive/leave the system at each node in each time slot.

We are now in position to present the following corollary to Theorem 3.1.

Corollary 5.1. *Consider the setup of problem $\mathcal{D}$ with $X \subseteq \operatorname{conv}(Y)$ and the following updates:*

$$x_k \in \arg\min_{x \in X} L(x, \lambda_k) \tag{5.9}$$

$$\lambda_{k+1} = [\lambda_k + Ax_k - B_k]^+, \tag{5.10}$$

where B_k is a bounded and i.i.d. random variable with mean b. Suppose the arrival rate into the system is in the interior of the capacity region, that is, there exists a point $x \in X$ such that $Ax - b \prec 0$. Then, we have that

$$\lim_{k\to\infty} \mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k} \lambda_i\right) \prec \infty. \tag{5.11}$$

Proof. The corollary follows directly from claim (iv) in Theorem 3.1, but we prove it in order to provide a simplified version of the theorem for feasibility problems.

First of all observe that $\mathbf{E}(\|[\lambda_{k+1}]^+\|_2^2 \mid \lambda_k) = \mathbf{E}(\|[\lambda_k + Ax_k - B_k]^+\|_2^2 \mid \lambda_k) \leq \mathbf{E}(\|\lambda_k + Ax_k - B_k\|_2^2 \mid \lambda_k) = \|\lambda_k\|_2^2 + \mathbf{E}(\|Ax_k - B_k\|_2^2 \mid \lambda_k) + 2\mathbf{E}(\lambda_k^T(Ax_k - B_k) \mid \lambda_k)$ where the expectation is taken with respect to B_k. Rearranging terms we can write

$$\mathbf{E}(\|\lambda_{k+1}\|_2^2 \mid \lambda_k) - \|\lambda_k\|_2^2 \leq \mathbf{E}(\|Ax_k - B_k\|_2^2 \mid \lambda_k) + 2\mathbf{E}(\lambda_k^T(Ax_k - B_k) \mid \lambda_k).$$

Now, since X is a bounded set and B_k a bounded random variable we have that $\max_{x\in X} \|Ax_k - B_k\|_2^2 := \sigma_g^2$ is finite, and therefore

$$\begin{aligned}\mathbf{E}(\|\lambda_{k+1}\|_2^2 \mid \lambda_k) - \|\lambda_k\|_2^2 &\le \sigma_g^2 + 2\mathbf{E}(\lambda_k^T(Ax_k - B_k) \mid \lambda_k)\\ &= \sigma_g^2 + 2\lambda_k^T(Ax_k - b),\end{aligned} \tag{5.12}$$

where the last step follows since λ_k and B_k are independent. We now proceed to upper bound the second term in the RHS of the last equation. First, observe that since $\operatorname{relint}(X_0)$ is non-empty by assumption, we have that for any vector $w \in \operatorname{relint}(X_0)$ there exists a constant $\gamma > 0$ such that $(Aw - b) \preceq -\gamma\mathbf{1}$. Now, observe that since $x_k \in \arg\min_{x\in X}\{\lambda_k^T(Ax - b)\}$ we have that $\lambda_k^T(Ax_k - b) \le \lambda_k^T(Aw - b)$, and therefore

$$\lambda_k^T(Ax_k - b) \le -\gamma\lambda_k^T\mathbf{1} = -\gamma\|\lambda_k\|_1.$$

Using the last equation in (5.12) and rearranging terms it follows that $2\gamma\|\lambda_k\|_1 \le \sigma_g^2 + \|\lambda_k\|_2^2 - \mathbf{E}(\|\lambda_{k+1}\|_2^2 \mid \lambda_k)$. Taking expectations with respect to B_i, $i = 1, \dots, k$, and using the fact that B_i and λ_i are indpendent

$$2\gamma\mathbf{E}(\|\lambda_k\|_1) \le \sigma_g^2 + \mathbf{E}(\|\lambda_k\|_2^2) - \mathbf{E}(\|\lambda_{k+1}\|_2^2) \tag{5.13}$$

Applying the expansion of λ_k recursively yields

$$2\gamma\sum_{i=1}^{k}\mathbf{E}(\|\lambda_k\|_1) \le \sigma_g^2 k + \|\lambda_1\|_2^2 - \mathbf{E}(\|\lambda_{k+1}\|_2^2). \tag{5.14}$$

Dividing both sides by $2\gamma k$ and dropping $\mathbf{E}(\|\lambda_{k+1}\|_2^2)$ because it is non-negative

$$\frac{1}{k}\sum_{i=1}^{k}\mathbf{E}(\|\lambda_i\|_1) \le \frac{\sigma_g^2}{2\gamma} + \frac{\|\lambda_1\|_2^2}{2\gamma k}. \tag{5.15}$$

Finally, since $\lambda(j) \le \|\lambda\|_1$ for all $j = 1, \dots, n$ we have that $\lim_{k\to\infty}\mathbf{E}(\frac{1}{k}\sum_{i=1}^{k}\lambda_i(j)) \le \sigma_g/(2\gamma) < \infty$, which concludes the proof. ■

Corollary 5.1 says that when the mean vehicle arrival/departure rate is in the interior of the capacity region then the expected value of the average of the Lagrange multipliers is bounded. Note the similarities between (5.11) and (5.1) in the definition of system stability. However, λ_k here is not a queue.

Now consider the special case where $X = \operatorname{conv}(Y)$, and observe that since (5.7) is a linear programme and X a polytope, for every $\lambda \in \mathbf{R}_+^n$ we have that

$$\arg\min_{y\in Y} L(y, \lambda) \subseteq \arg\min_{x\in X} L(x, \lambda). \tag{5.16}$$

That is, there exists a point $y \in Y$ that minimises $L(\cdot, \lambda)$ for every $\lambda \in \mathbf{R}^n_+$, and therefore we can use a value $y \in Y$ in update (5.5) to obtain

$$\lambda_{k+1} = [\lambda_k + Ay_k - B_k]^+. \tag{5.17}$$

Since all the elements in update (5.17) are discrete, we can clearly identify a Lagrange multiplier with a real queue, *i.e.* $\lambda_k = Q_k$. Hence, if the mean vehicle arrival/departure rate is in the interior of the capacity region, by Corollary 5.11 we have that

$$\lim_{k\to\infty} \mathbf{E}\left(\frac{1}{k}\sum_{i=1}^{k} Q_i\right) \prec \infty \tag{5.18}$$

and so the system is stable.

Note that the choice of discrete action used in (5.17) depends exclusively on vector Q_k, and not on other factors such as network constraints. However, we can have flexibility on how to select discrete actions by using results in Chapter 4. In short, we now suppose $X \subseteq \text{conv}(Y)$ and consider updates

$$\lambda_{k+1} = [\lambda_k + Ax_k - B_k]^+, \tag{5.19}$$

$$Q_{k+1} = [Q_k + Ay_k - B_k]^+, \tag{5.20}$$

where $\lambda_1 = Q_1 \succeq 0$. Recall that if we construct a sequence of discrete actions $\{y_k\}$ such that $\|\sum_{i=1}^{k}(x_i - y_i)\|_2 \le \epsilon$ we then have

$$\|\lambda_k - Q_k\|_2 \le 2\epsilon\|A\|_2 \qquad k = 1, 2, \dots \tag{5.21}$$

Next, we present an algorithm that is based on Theorem 4.1 and update (4.10) that allows us to choose the order in which the discrete actions are made. As explained in Chapter 4, for the construction of such sequences we need to define

$$W := [y_1, \dots, y_{|Y|}], \tag{5.22}$$

$$E := \{v_1, \dots, v_{|Y|}\}, \tag{5.23}$$

$$U := \text{conv}(E), \tag{5.24}$$

where v_j is an $|Y|$-dimensional standard basis vector, *i.e.* all elements of vector v_j are equal to 0 except the j'th element that it is equal to 1.

Algorithm 1 selects actions in blocks of T_c elements, which in a traffic control context can be regarded as the duration of a traffic light cycle. A particular characteristic of the algorithm is that there are no restrictions on the order in which actions must appear, and therefore the actions in a block can be reordered in order to capture specific problem

---

Define parameters:

$T \in \mathbf{N}$;
$T_c \in T|Y|$; *# block length*
$\delta = 0$;
Let $\hat{E} = [\hat{e}(1), \ldots, \hat{e}(T_c)]$ where $\hat{e}(j) \in E$, $j = 1 \ldots, T_c$.

At each time slot k **do:**

(i) Obtain vector u_k solving convex optimisation

$$u_k \in \arg\min_{u \in U} \|Wu - x_k\|_2^2$$

$$\delta = \delta + u_k$$

(ii) At the end of every interval of T_c slots (*i.e.* if $\mathrm{mod}(k, T_c) = 0$) do:

for $j = 1, \ldots, T_c$ **do**
  $\hat{e}(j) \in \arg\min_{e \in E} \|\delta - e\|_\infty$;
  $\delta = \delta - \hat{e}(j)$;
end
Reorder columns of matrix $\hat{E}$; *# optional*

(iii) Recover discrete action:

$$j = \mathrm{mod}(k, T_c)$$

$$y_k = W\hat{e}(j)$$

Algorithm 1: Block Algorithm

characteristics (*i.e.* meta-actions). We show how this algorithm is useful in practice in the following example.

5.1.3. Numerical Example. We consider a simple example in which the average rate of vehicles that can go through an intersection depends on the order in which traffic signal control actions are taken. Consider the intersection shown in Figure 5.1.2, but for simplicity we do not allow right or left turns in the intersection (*i.e.* links 5, 6, 7 and 8 do not exist). The network incidence matrix in the intersection is given by

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}^T. \tag{5.25}$$

Time is divided into slots of duration 2 seconds, and in each time slot the network controller activates one of the phases in the intersection.[5] Phase 1 gives the right of way to queue

[5]Since there is only one intersection in the network activating a phase in the intersection is equivalent to selecting a network action.

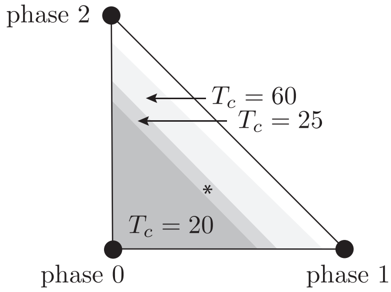


FIGURE 5.1.4. Schematic illustration of the intersection capacity region. The asterisk in the figure indicates the mean vehicle arrival rate to queues 1, 2, 3 and 4. Phase 1 corresponds to giving right of way to queue 1 and 2; and phase 2 to giving right of way to queue 3 and 4.

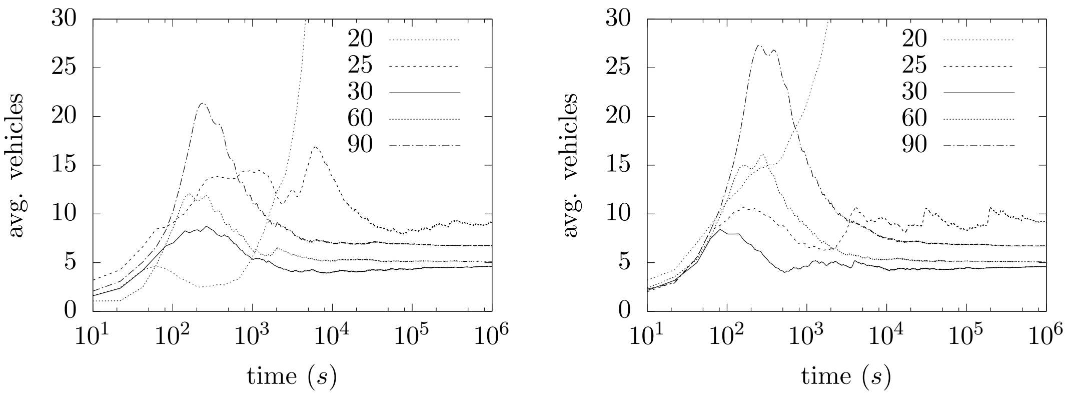
1 and 2, which corresponds to activating links 1 and 2; and phase 2 gives right of way to queue 3 and 4, which corresponds to activating links 3 and 4. We also include a phase 0 or "null action", which corresponds to not activating any of the links. The action set in this example is given by

$$Y = \{y_0, y_1, y_2\}, \tag{5.26}$$

where $y_0 = [0, 0, 0, 0]^T$, $y_1 = [1, 1, 0, 0]^T$, $y_2 = [0, 0, 1, 1]^T$ correspond, respectively, to phase 0, 1 and 2.

We will assume that exactly one vehicle moves from one queue to another in a time slot, and to capture the overhead of changing between phase 1 and 2 (start-up and clearance loss times) we will add constraints on the order in which actions can be selected by the network controller. In particular, the network controller cannot change between phase 1 and phase 2 without selecting phase 0 in $T_d \in \mathbf{N}$ consecutive slots. For example, if $T_d = 2$ then $\{y_0, y_0, y_1, \ldots, y_1, y_0, y_0, y_2, y_2, \ldots\}$ is a valid sequence of control actions.

The network capacity depends on the order in which control actions are selected. To characterise the network capacity region observe that if phase 1 and 2 *both* appear in an interval of T_c slots, then, phase 0 should appear in at *least* $2T_d$ time slots. However, if phase 1 and 2 are selected consecutively in an interval, then, phase 0 would only need to appear in *exactly* $2T_d$ time slots (which maximises the vehicle rate through the intersection). Hence, the network capacity region is a function of T_c and T_d, and it is given by $X = \eta\,\mathrm{conv}(Y)$ where $\eta = 1 - 2T_d/T_c$. Figure 5.1.4 shows, schematically, the network capacity region as a function of T_c. Observe from the figure that X is a subset of $\mathrm{conv}(Y)$, and that the network capacity region increases as T_c gets larger, *i.e.* the impact of the start-up and clearance loss time overheads is reduced.



(A) Queue 1 (B) Queue 3

FIGURE 5.1.5. Illustrating the average queue length (*i.e.* average number of vehicles) of queues 1 and 3 as a function of time for different values of $T_c \in \{20, 25, 30, 60, 90\}$; queues 2 and 4 have, respectively, similar behaviour than queue 1 and 3.

Simulation. We run a simulation (*i.e.* updates (5.9) and (5.10)) with

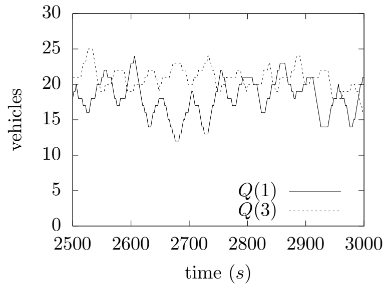
$$B_k = [-\tilde{B}_k(1), -\tilde{B}_k(2), -\tilde{B}_k(3), -\tilde{B}_k(4), 1, 1, 1, 1]$$

where $\tilde{B}_k(j),\ j = 1, \ldots, 4$ are i.i.d. Bernoulli random variables taking values in $\{0, 1\}$ with mean 0.44, 0.44, 0.22, 0.22. In all simulations we use initial condition $\lambda_1 = Q_1 = 0$, and $T_d = 4$ slots. Traffic control actions y_k are selected using Algorithm 1. Actions are reordered within a block of T_c time slots so that phase 1 and 2 appear consecutively, but selecting first phase 0 for a total of T_d slots.

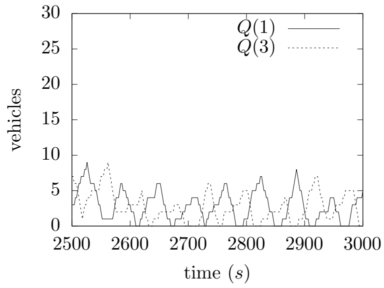
Figure 5.1.5 shows the evolution of the average of the queue lengths for queue 1 and queue 3 for $T_c \in \{20, 25, 30, 60, 90\}$. When $T_c = 20$ slots, the set of arrival rates lies outside of the capacity region (see Figure 5.1.4), and as a result the occupancies of the queues in the system grow unbounded. When $T_c \in \{25, 30, 60, 90\}$, the mean vehicle arrival rate is in the interior of the network capacity region and our optimisation (Theorem 5.1 and Algorithm 1) stabilises the system. Observe from Figure 5.1.5 that the averages of the queue occupancies remain bounded.

An important observation is that the average queue occupancy depends on T_c, however, there is not a direct relationship. Observe from Figure 5.1.5 that the largest average queue occupancy happens when $T_c = 25$, the lowest when $T_c = 30$, and when $T_c \in \{60, 90\}$ the average queue occupancies lie in between. When $T_c = 25$, the large average queue occupancy is a result of the mean arrival rate in the system being close to the boundary of the network capacity region. In contrast, when $T_c = 90$ the large average queue occupancies are a consequence of how actions are reordered in Algorithm 1. Namely, phases/actions are selected consecutively independently of whether there are vehicles to be served. Figure 5.1.6 shows the occupancies of queue 1 and 3 for a time period of 500 slots. Observe that when $T_c = 25$ slots the queues are never empty, but as T_c increases, the queues tend to get larger and are also empty for longer intervals of times. This kind of behaviour will affect not just the mean but also the distribution of the vehicle waiting times in the intersection. This can be observed in Figure 5.1.7, which shows the delay distribution of queue 1 and queue 3 for different values of T_c. The impact of T_c on the delay distribution is important if, for example, we want to provide guarantees on the maximum amount of time that a vehicle will have to wait in an intersection. Similarly, since the delay distribution is related to the size of a queue, we have that T_c will play a fundamental role in preventing deadlocks. In this example, the optimal value of T_c is 30, but in real networks T_c will need to be adjusted dynamically in order to obtain both, stability and good delay performance.

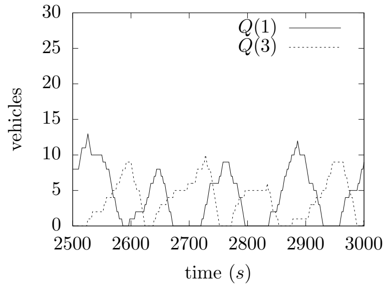
We could also design an algorithm based on Theorem 4.3 where actions are selected in order to keep γ_k bounded. For example, a possible strategy would be to change phase when either (i) one queue in an active phase is empty, or (ii) when both queues in an active phase are empty—and so avoiding selecting phases in which there are no vehicles waiting to be served. We do not characterise the network capacity region of the algorithms and just evaluate them experimentally. We run a simulation using the algorithms and show the results in Figures 5.1.8 and 5.1.9. Observe that now there is a notable difference between the treatment that flows have, in particular, the occupancy of queue 1 is significantly larger than the occupancy of queue 3; and consequently, the delay distribution of the vehicles waiting times in queue 3 is drastically affected. The results suggest that these traffic policies are not fair, and that making traffic control decisions based only on queue occupancies may not be enough if we want to capture metrics apart from the stability of the system. This fairness issue could be tackled by either adding an utility function in the optimisation, or by designing scheduling policies directly that consider other parameters apart from the occupancy of queues.



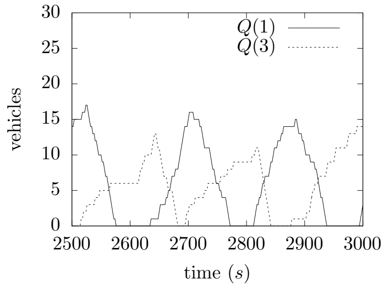
(A) $T_c = 25$



(B) $T_c = 30$

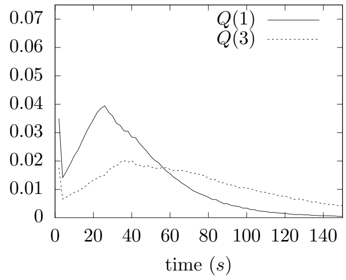


(C) $T_c = 60$

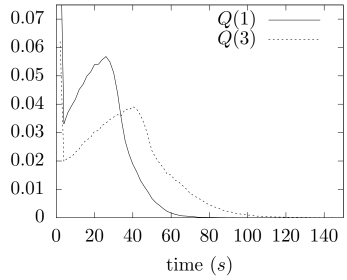


(D) $T_c = 90$

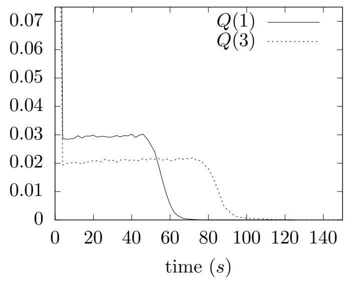
FIGURE 5.1.6. Illustrating the queue occupancies depending on T_c. The start of phase 1 and 2 can be identified, respectively, by the peaks of queue 1 and 3.



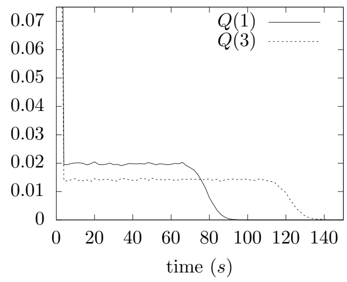
(A) $T_c = 25$.



(B) $T_c = 30$.



(C) $T_c = 60$.



(D) $T_c = 90$.

FIGURE 5.1.7. Illustrating the distribution of the waiting times (delay) depending on T_c.

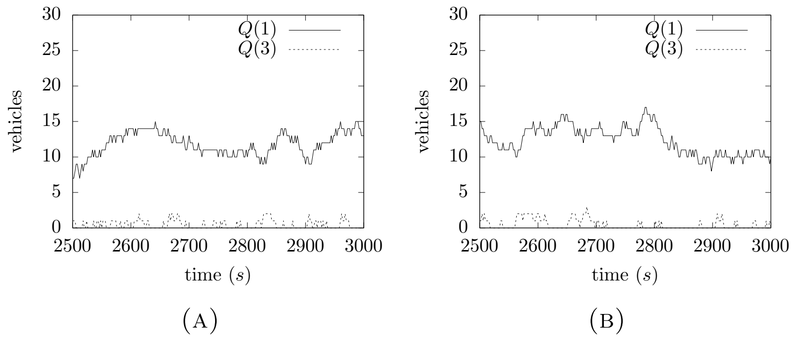


FIGURE 5.1.8. Illustrating the queue sizes when a phase change is performed when a queue in an active phase is empty (5.1.8a), or when both queues in an active phase are empty (5.1.8b).

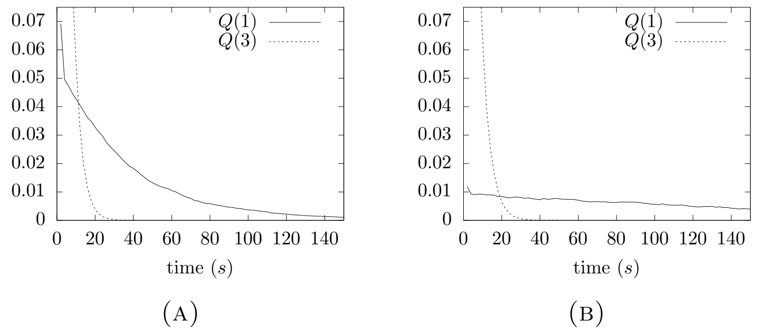


FIGURE 5.1.9. Illustrating the delay distribution of $Q(1)$ and $Q(3)$ when a phase change is performed when a queue in an active phase is empty (5.1.8a), or when both queues in an active phase are empty (5.1.8b).

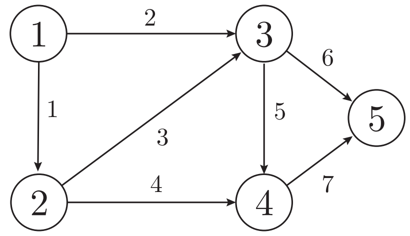


FIGURE 5.2.1. Illustrating the network of Example 5.2.

5.2. Distributed and Asynchronous Packets Transmissions

We now present a classic network flow problem and show that (i) scaled queue occupancies can be used as surrogates for the Lagrange multipliers in the subgradient updates, (ii) that the optimisation does not require the statistics of the arrival process in the network to be known in advance, and (iii) nodes make decisions as to whether to transmit packet or not in a distributed and asynchronous manner.

5.2.1. Problem Setup.

Consider the network illustrated in Figure 5.2.1 with $m = 5$ nodes and $n = 7$ links.[6] Packets arrive in the system at nodes 1 and 2, and they must be transmitted through the network until they reach node 5, where the packets will leave the system. The incidence matrix of the network is given by

$$A = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The time in the system is divided in slots of equal duration, and each time slot corresponds to transmitting a fraction of a packet. For simplicity, we will assume that all the packets in the network have a duration of 10 slots, and so an action y_k corresponds to transmitting a tenth of a packet. The network action set is $Y = \prod_{j=1}^{n}\{0, 1\}$ and so nodes can select a discrete action in each time slot independently of the discrete actions selected by the other nodes. The goal of the problem is to design a distributed scheduling policy that minimises the cost of allocating data to links—for example, suppose that a network operator charges for a link usage.

[6]The network is taken from the flow example in [BXMM07]

Fluid Formulation. The convex formulation of the problem is

$$\begin{aligned} \underset{x \in X}{\text{minimise}} \quad & f(x) = \textstyle\sum_{j=1}^{n} f_j(x(j)) \\ \text{subject to} \quad & Ax + b \preceq 0 \end{aligned} \tag{5.27}$$

where $X := \prod_{j=1}^{n} X_j$, $X_j := \text{conv}(Y_j)$, $Y_j = \{0,1\}$ for all $j = 1,\dots,n$, $b \in \mathbf{R}^m$ is a vector containing the mean packet arrival/departure rate in the network at each of the nodes, and $f_j : \mathbf{R} \to \mathbf{R}$ are convex functions that capture the cost of using each link $j = 1,\dots,n$.

Optimisation (5.27) can be solved with the dual subgradient method, and since the objective function is fully separable and constraints are linear, the computation of a (dual) subgradient can be carried out in a distributed manner. In particular, in each iteration we have updates

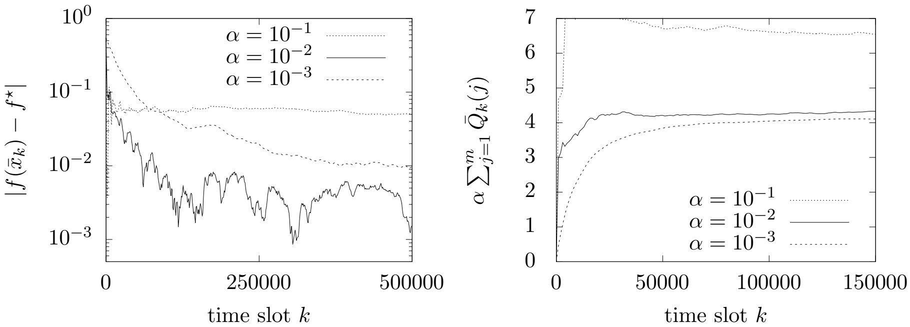
$$x_k(j) \in \arg\min_{x \in X_j} \{f_j(x) + \alpha r_k(j) x\}, \qquad j = 1,\dots,n \tag{5.28}$$

$$Q_{k+1} = [Q_k + Ay_k + B_k]^+, \tag{5.29}$$

where $y_k \in Y$, $r_k = Q_k^T A$, and $Q_k \in \mathbf{R}^m$ is a vector of queue occupancies, and $B_k \in \mathbf{R}^m$ is a random variable that captures the *packet* arrival/departure from the system in each of the nodes. Recall that in order to use Theorem 3.1 we need αQ_k to be an approximate Lagrange multiplier; that strong duality holds in the fluid problem; and that $\mathbf{E}(B_k) = b$ for all k. Interesting observations from updates (5.28)-(5.29) are that we never compute the Lagrange multipliers in the optimisation since using approximate Lagrange multipliers is enough, and that update (5.28) does not require us to know the mean packet arrival/departure rate of packets in the system.

5.2.2. Simulation. We run updates (5.28) and (5.29) with $f_j(x(j)) = x(j)^2$ for every link $j = 1,\dots,n$, $b = [0.02, 0.06, 0, 0, -0.1]^T$, $B_k(j)$, $j = 1,2$ are Bernoulli with $\mathbf{E}(B_k(j)) = b(j)$, and $B_k(j)$ for $j = 3,4,5$ are equal to $b(j)$ for all k, *i.e.* nodes 3 and 4 do not receive exogenous packets, and the service of node 5 is deterministic. The sequence $\{y_k\}$ of discrete actions is obtained with Corollary 4.3 and $\bar{\tau} = 10$, *i.e.* the transmission of a packet (meta-action) takes exactly 10 time slots and cannot be interrupted.

Figure 5.2.2 shows the convergence of $f(\bar{x}_k)$ to a ball around the optimum for $\alpha = \{10^{-1}, 10^{-2}, 10^{-3}\}$, and how the average of the (scaled) queue occupancies remains bounded. Observe from Figure 5.2.2a that the convergence of the utility improves as parameter α is reduced, however, the convergence time increases as α gets smaller. The latter can be clearly seen in Figure 5.2.2b. Observe from the figure that the sum of the scaled queue occupancies converge to a ball around $\mathbf{1}^T \lambda^\star(\delta) = 4.25$, and that the size of the ball depends on parameter α. Also, note that since a queue occupancy is inversely proportional to step



(A) (B)

FIGURE 5.2.2. Illustrating the convergence of the utility function and the (scaled) mean queue occupancies in Example 5.2.

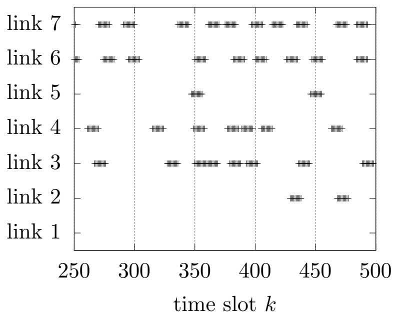


FIGURE 5.2.3. Illustrating the packet transmissions for each link in the network for an interval of 250 time slots.

size α, we obtain the typical tradeoff between optimality and delay. Figure 5.2.3 shows the link's usage for an interval of 250 time slots. Observe from the figure that all the packet transmissions have a duration of 10 slots, and that they are asynchronous, *i.e.* a node can start a new packet transmission in any slot without coordinating with the other nodes.

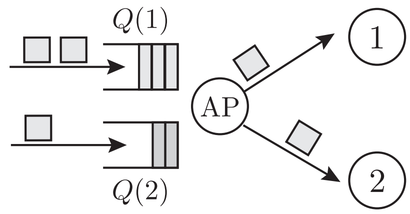


FIGURE 5.3.1. Illustrating the network in the example of Section 5.3. The Access Point (AP) sends packets from $Q(1)$ to node 1, and from $Q(2)$ to node 2.

5.3. Packet Transmissions with Constraints

This problem is in spirit like the traffic signal control example, but we put it into a networking context and consider an utility function.

Consider the network shown in Figure 5.3.1, an Access Point (AP) that transmits to two wireless nodes. Time is slotted and in each time slot packets arrive at the queues of the AP, $Q(1)$ and $Q(2)$. In each time slot the AP takes an action from action set $Y :=$ $\{y(0), y(1), y(2)\} = \{[0,0]^T, [1,0]^T, [0,1]^T\}$, where each action corresponds, respectively, to not transmitting, to transmitting one packet from $Q(1)$ to node 1, and to transmitting one packet from $Q(2)$ to node 2.

The transmission protocol of the AP has constraints on how actions can be selected. In particular, it is not possible to select action $y(1)$ after $y(2)$ without first selecting $y(0)$. In the same way, it is not possible to select $y(2)$ after $y(1)$ without first selecting $y(0)$. However, $y(1)$ or $y(2)$ can be selected consecutively. An example of an admissible sequence is

$$\{y(1), y(1), y(1), y(0), y(2), y(0), y(1), y(1), y(0), y(2), y(2), \dots\}.$$

These type of constraints appear in different areas, and are usually known in the literature as reconfiguration or switchover delays [ÇM15]. In this example, such constraints on how to select actions might correspond to asking for the Channel State Information (CSI) in order to adjust the transmission parameters.[7]

Our goal is to design a scheduling policy for the AP (select actions from set Y) in order to minimise a convex cost function of the average throughput $\bar{x}_k$, and ensure that the system is stable, *i.e.* the queues do not overflow and so all traffic can be served.

[7]The CSI in wireless communications is in practice requested periodically, and not only at the beginning of a transmission, but we will assume this for simplicity. The extension is nevertheless straightforward.

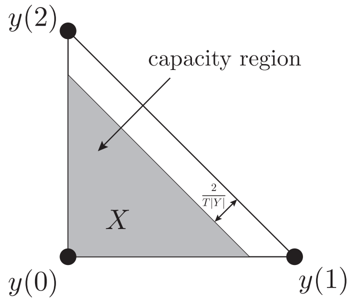


FIGURE 5.3.2. Network capacity region of the example in Section 5.3. $X = \eta\,\mathrm{conv}(Y)$ with $\eta = 1 - 2/(T|Y|)$.

5.3.1. Problem Setup.

Fluid Formulation: The convex or fluid formulation of the problem is

$$\begin{array}{ll} \underset{x \in X}{\text{minimise}} & f(x) \\ \text{subject to} & b \preceq x \end{array}$$

where $f : \mathbf{R}^2 \to \mathbf{R}$, $b \in \mathbf{R}^2_+$ and X is a bounded convex subset from $\mathrm{conv}(Y) \subset \mathbf{R}^2$ that depends on the protocol constraints, *i.e.* on how actions can be selected. If there were no constraints on the order in which actions could be selected we would have $X := \mathrm{conv}(Y)$, however, this is *not* the case. Nonetheless, characterising X is as simple as noting that if we have a subsequence of actions of length $T|Y|$, $T \in \mathbf{N}$ where $y(1)$ and $y(2)$ appear (each) consecutively, then $y(0)$ should appear *at least* twice in order to have a subsequence that is compliant with the transmission protocol. Conversely, any subsequence of length $T|Y|$ in which $y(0)$ appears at least twice can be reordered to obtain a subsequence that is compliant with the transmission protocol. Since from Section 4.3.1, we know we can always choose a subsequence of discrete actions and then reorder its elements, we just need to select set X such that $y(0)$ can be selected twice in a subsequence of $T|Y|$ elements. This will be the case when a point $x \in X$ needs to use action $y(0)$ at least $2/(T|Y|)$. Hence, we have that

$$X := \eta\,\mathrm{conv}(Y), \qquad (5.30)$$

where $\eta := 1 - 2/(T|Y|)$. Observe from Equation (5.30) that as T increases $\eta \to 1$ and therefore $X \to \mathrm{conv}(Y)$, *i.e.* the network capacity region increases. Figure 5.3.2 illustrates the network capacity region of the example.

Dual Subgradient Updates. At each time slot we have updates

$$x_k \in \arg\min_{x \in X} \{f(x) - \alpha Q_k^T x\}, \tag{5.31}$$

$$\lambda_{k+1} = [\lambda_k - x_k + B_k]^+, \tag{5.32}$$

$$Q_{k+1} = [Q_k - y_k + B_k]^+, \tag{5.33}$$

where $\lambda_k, Q_k \in \mathbf{R}^2_+$, B_k are i.i.d. random variables that take values $\{0, 1\}$ with mean $[b(1), b(2)]^T$. Action $y_k = We_k$, and e_k is obtained with (4.10) and the elements in a subsequences are reordered in order to meet the transmission protocol constraints.

5.3.2. Simulations. We run a simulation for 10000 iterations with objective function $f(x) = \|x\|_2^2$, $b = [0.25, 0.5]^T$, $\alpha \in \{5 \cdot 10^{-2}, 10^{-2}, 10^{-3}, 10^{-4}\}$, $\lambda_1 = \alpha Q_1 = 0$, $T = 3$ (so the number of elements in a subsequence is 9 since $|Y| = 3$). Figure 5.3.3 shows the convergence of the Lagrange multipliers to a ball around $\lambda^\star = [0.5, 1]^T$ for different step sizes. Note that the size of the ball depends on the step size used, and that for $\alpha = 10^{-4}$ the Lagrange multipliers do not have time to reach the ball in 10000 iterations. The slow convergence of the Lagrange multipliers also affects the convergence of the objective function. Observe in Figure 5.3.4 that this is actually the case, and that surprisingly, with step size $\alpha \in \{5 \cdot 10^{-2}, 10^{-2}\}$ we obtain a relatively fast convergence with moderate accuracy.

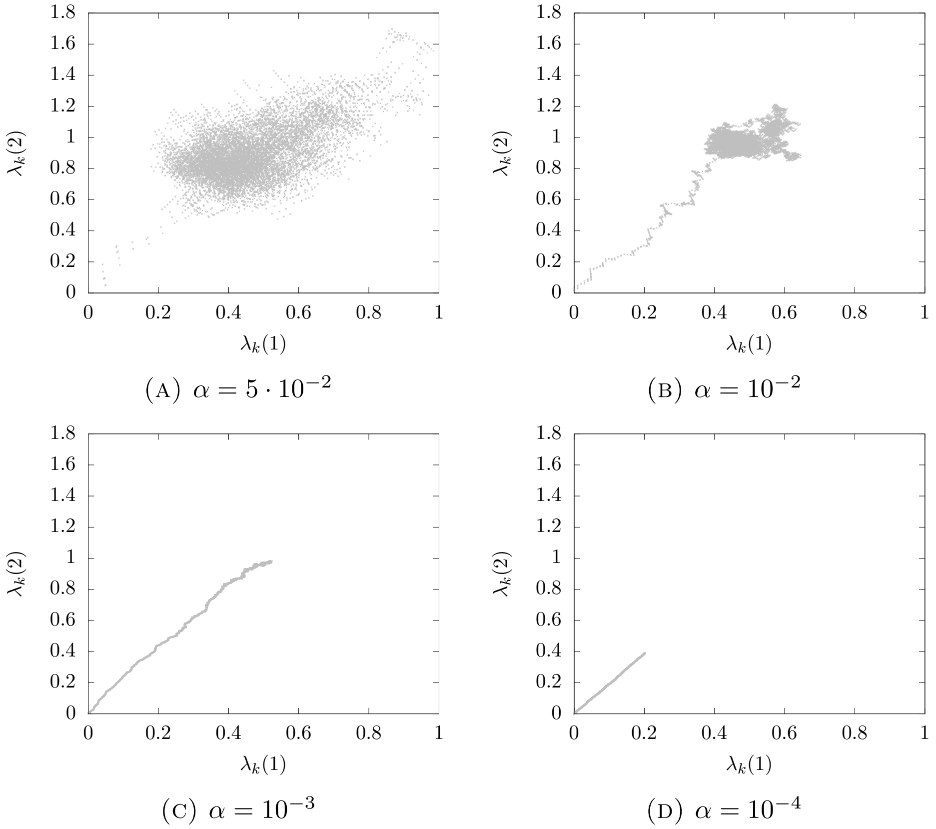


FIGURE 5.3.3. Illustrating the convergence of λ_k to a ball around $\lambda^\star = [0.5, 1]^T$.

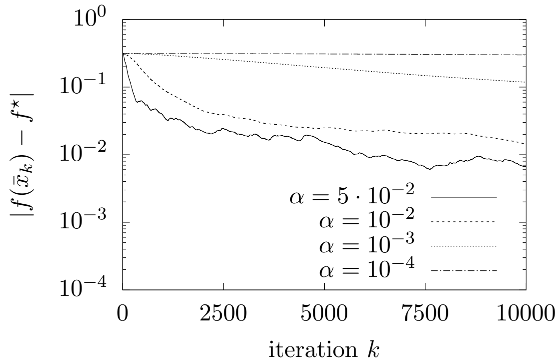


FIGURE 5.3.4. Illustrating the difference $|f(\bar{x}_k) - f^\star|$ for different step sizes.

CHAPTER 6

Conclusions

In this thesis we have studied how max-weight can be encompassed within the body of convex optimisation, and shown it is a special case of the stochastic dual subgradient method with ϵ_k subgradients. Our analysis clarifies the fundamental properties required for convergence, and brings to the fore the use of ϵ_k-subgradients as a key component for modelling problem characteristics, including discrete actions. One of the great advantages of our approach is that scheduling policies can be decoupled from the choice of convex optimisation algorithm or subgradient used to solve the dual problem. As a result, it is possible to design scheduling policies with a high degree of flexibility. Other consequences of our approach are that scaled queue occupancies can be used directly as surrogates for the Lagrange multipliers in the optimisation, and that it is possible to obtain bounds that are not asymptotic.

The work in this thesis brings clarity and a fresh outlook to network optimisation problems, but also makes max-weight accessible from a convex optimisation perspective and therefore to a wider audience beyond networking and control. We believe this will lead to new applications currently not covered by either max-weight approaches or convex optimisation. We would also like to emphasise that we have focused on max-weight's main features (discrete actions and optimal decisions without previous knowledge of the underlying randomness in the system) because of their importance in the literature. However, the perturbations in the optimisation framework can be used to model a wider range of problems' features. An example of this is the asynchronous dual updates presented in Section 4.2, where approximate Lagrange multipliers are chosen in each iteration in order to model a specific kind of behaviour.

Natural extensions to this work include weakening the requirements on the perturbations ϵ_k and δ_k in the optimisation framework, and applications that further illustrate the power of the results.

APPENDIX A

Lagrange Dual Function

We show how to obtain the Lagrange dual function using conjugate duality. We start by recalling the definition of the (convex) conjugate function.

DEFINITION A.1 (Convex Conjugate). *For any function* $\psi : \mathbf{R}^n \to \mathbf{R}$*, the function* $\psi^* : \mathbf{R}^n \to \mathbf{R}$ *defined by*

$$\psi^*(y) := \sup_{x \in \mathbf{R}^n} \{y^T x - \psi(x)\}$$

is the convex conjugate of ψ*, and*

$$\psi^{**}(x) = \sup_{y \in \mathbf{R}^n} \{x^T y - \psi^*(y)\}$$

is the biconjugate of ψ*.*

One the important properties of the conjugate function is that it is always convex, even though ψ is not. Further, when ψ is convex, proper and lower semi continuous we have that $\psi = \psi^{**}$, and otherwise $\psi \geq \psi^{**}$. The conjugate function can be thought of as representing a convex function in a different form, and one can take the conjugate of the conjugate to recover the original function.

The Lagrange dual function arises through the partial dualisation of a problem for a specific choice of perturbation. Similar ideas can be found in [ET99, Chapter 3] and [RW98, Chapter 11]. Consider optimisation problem

$$\begin{aligned} \underset{x \in X}{\text{minimise}} \quad & f(x) \\ \text{subject to} \quad & g(x) \preceq 0 \end{aligned}$$

where X is a convex set but not necessarily bounded. We use inequality constraints for simplicity, but similar steps can be used for equality constraints. Define

$$l(\delta) := \inf_{x \in X} \{f(x) + \mathbf{I}(g(x) + \delta)\} \tag{A.1}$$

where $\delta \in \mathbf{R}^m$ is a perturbation on the constraints, and $\mathbf{I}$ is the indicator function, *i.e.*

$$l(\delta) = \begin{cases} f(x) & g(x) + \delta \preceq 0 \\ +\infty & \text{otherwise.} \end{cases} \tag{A.2}$$

Observe that l might not be continuous but it is still convex in δ. We can write the convex conjugate of l with respect to perturbation δ as follows

$$
\begin{aligned}
l^*(\lambda) &= \sup_{\delta \in \mathbf{R}^m} \{\lambda^T \delta - \inf_{x \in X} l(x,\delta)\}, \\
&= \sup_{\delta \in \mathbf{R}^m} \{\lambda^T \delta - \inf_{x \in X} \{f(x) + \mathbf{I}(g(x)+\delta)\}\}, \\
&= \sup_{\delta \in \mathbf{R}^m} \{\lambda^T \delta + \sup_{x \in X} \{-f(x) - \mathbf{I}(g(x)+\delta)\}\}, \\
&= \sup_{\substack{x \in X,\ \delta \in \mathbf{R}^m, \\ g(x)+\delta \preceq 0}} \{\lambda^T \delta - f(x)\}
\end{aligned} \tag{A.3}
$$

where the last equation follows from the fact that $-\mathbf{I}(g(x)+\delta) = -\infty$ if $g(x)+\delta \succ 0$. Note we can always do that since for any $x \in X$ there always exists $\delta \in \mathbf{R}^m$ such that $g(x)+\delta \preceq 0$. Next, we can write (A.3) as

$$l^*(\lambda) = \sup_{\substack{x \in X,\ \delta \in \mathbf{R}^m, \\ \delta \preceq -g(x)}} \{\lambda^T \delta - f(x)\}$$

Now, observe that since we require that $\delta(j) \preceq -g_j(x)$, if $\lambda(j) < 0$ for some j then $\delta(j)$ will be selected equal to $-\infty$ and therefore $l^*(\delta) = \infty$. Also, if $\lambda(j) \geq 0$ then δ will be chosen as large as possible, *i.e.* $\delta(j) = -g_j(x)$ independently of the $x \in X$. Hence,

$$l^*(\lambda) = \begin{cases} \sup_{x \in X} \{-f(x) - \lambda^T g(x)\} & \lambda \succeq 0 \\ +\infty & \lambda \prec 0 \end{cases} \tag{A.4}$$

Importantly, we are just having another representation of our original function $l(\delta)$, we are not changing the optimisation problem. In fact, we can solve the original problem by minimising $l^*(\lambda)$ for $\lambda \succeq 0$, or equivalently

$$\sup_{\lambda \succeq 0} \ -l^*(\lambda). \tag{A.5}$$

It is enlightening to see that

$$h(\lambda) := -l^*(\lambda) = \begin{cases} \inf_{x \in X} \{f(x) + \lambda^T g(x)\} & \lambda \succeq 0 \\ -\infty & \lambda \prec 0 \end{cases}$$

i.e. we have obtained the Lagrange dual function. From the latter one can easily obtain

$$\sup_{\lambda \succeq 0} \inf_{x \in X} \{f(x) + \lambda^T g(x)\} \leq \inf_{x \in X} \sup_{\lambda \succeq 0} \{f(x) + \lambda^T g(x)\}$$

where equality will hold when the Slater condition is satisfied—which is one of the conditions in Fenchel's duality theorem [Roc97, Theorem 31.1].

Bibliography

[AD06] J. W. Antonis Dimakis. Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits. *Advances in Applied Probability*, 38(2):505–521, 2006.

[BB87] D. P. Bertsekas and D. E. Baz. Distributed asynchronous relaxation methods for convex network flow problems. *SIAM Journal on Control and Optimization*, 25(1):74–85, 1987.

[BC12] T. Bonald and D. Cuda. Rateoptimal scheduling schemes for asynchronous inputqueued packet switches. *SIGMETRICS Perform. Eval. Rev.*, 40(3):95–97, January 2012.

[Ber60] B. Bernstein. Proof of caratheodory's local theorem and its global application to thermostatics. *Journal of Mathematical Physics*, 1(3):222–224, 1960.

[Ber98] D. P. Bertsekas. *Network optimization: continuous and discrete models*. Citeseer, 1998.

[Ber99] D. P. Bertsekas. *Nonlinear Programming: Second Edition*. Athena Scientific, 1999.

[BNO03] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex analysis and optimization*. Athena Scientific, 2003.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[BT89] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989.

[BT03] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 5 2003.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BXMM07] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods, 2007.

[CKLS08] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Transactions on Information Theory*, 54(2):572–594, Feb 2008.

[ÇLM12] G. D. Çelik, L. B. Le, and E. Modiano. Dynamic server allocation over time-varying channels with switchover delay. *IEEE Transactions on Information Theory*, 58(9):5856–5877, Sept 2012.

[ÇM15] G. D. Çelik and E. Modiano. Scheduling in networks with time-varying channels and reconfiguration delay. *IEEE/ACM Transactions on Networking*, 23(1):99–113, Feb 2015.

[CNS14] T. H. Chang, A. Nedić, and A. Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, 59(6):1524–1538, June 2014.

[DHS11] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[Die13] J. Dieudonné. *Foundations of modern analysis*. Read Books Ltd, 2013.

[Erm66] Y. M. Ermoliev. Methods of solution of nonlinear extremal problems. *Cybernetics*, 2(4):1–14, 1966.

[ES05] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proceedings IEEE 24th Annual Joint*

Conference of the IEEE Computer and Communications Societies., volume 3, pp. 1794–1803. IEEE, 2005.

[ET99] I. Ekeland and R. Témam. *Convex Analysis and Variational Problems.* Society for Industrial and Applied Mathematics, 1999.

[FKM05] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pp. 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[Gar83] N. H. Gartner. Opac: A demand-responsive strategy for traffic signal control. In *62nd Annual Meeting of the Transportation Research Board*, 1983.

[GFDLFW13] J. Gregoire, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn. Back-pressure Traffic Signal Control with Unknown Routing Rates. *arXiv preprint arXiv:1401.3357*, 2013.

[GNT06] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Optimization*, 1(1):1–144, 2006.

[GQF+15] J. Gregoire, X. Qian, E. Frazzoli, A. de La Fortelle, and T. Wongpiromsarn. Capacity-Aware Backpressure Traffic Signal Control. *Control of Network Systems, IEEE Transactions on*, 2(2):164–173, 2015.

[HN11] L. Huang and M. J. Neely. Delay reduction via lagrange multipliers in stochastic network optimization. *IEEE Transactions on Automatic Control*, 56(4):842–857, April 2011.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[HRBW81] P. Hunt, D. Robertson, R. Bretherton, and R. Winton. SCOOT - A Traffic Responsive Method of Coordinating Signals. Technical report, Transport and Road Research Laboratory, 1981.

[IGHT15] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas. A double-auction mechanism for mobile data-offloading markets. *IEEE/ACM Transactions on Networking*, 23(5):1634–1647, Oct 2015.

[KMT98] P. F. Kelly, K. A. Maulloo, and H. D. K. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[Kom88] H. Komiya. Elementary proof for sion's minimax theorem. *Kodai mathematical journal*, 11(1):5–7, 1988.

[KY14] F. Kelly and E. Yudovina. *Stochastic Networks.* Cambridge University Press, 2014.

[LBX11] B. Li, C. Boyaci, and Y. Xia. A refined performance characterization of longest-queue-first policy in wireless networks. *IEEE/ACM Transactions on Networking*, 19(5):1382–1395, Oct 2011.

[LL97] T. Larsson and Z. Liu. A lagrangean relaxation scheme for structured linear programs with application to multicommodity network flows. *Optimization*, 40(3):247–284, 1997.

[Low90] P. Lowrie. SCATS, Sydney Co-ordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic, 1990.

[LSS06] X. Lin, N. B. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, Aug 2006.

[Mar05] K. Marti. *Stochastic optimization methods.* Springer, 2005.

[Mey08] S. Meyn. *Control techniques for complex networks.* Cambridge University Press, 2008.

[Mey09] S. Meyn. Stability and asymptotic optimality of generalized maxweight policies. *SIAM Journal on Control and Optimization*, 47(6):3259–3294, 2009.

[MH01] P. Mirchandani and L. Head. A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.

[MHS14] S. T. Maguluri, B. Hajek, and R. Srikant. The stability of longest-queue-first scheduling with variable packet sizes. *IEEE Transactions on Automatic Control*, 59(8):2295–2300, Aug 2014.

[MMT14] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis. Max-weight scheduling in queueing networks with heavy-tailed traffic. *IEEE/ACM Transactions on Networking*, 22(1):257–270, Feb 2014.

[MMT16] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis. Delay stability of back-pressure policies in the presence of heavy-tailed traffic. *IEEE/ACM Transactions on Networking*, 24(4):2046–2059, Aug 2016.

[MS14] S. T. Maguluri and R. Srikant. Scheduling jobs with unknown duration in clouds. *IEEE/ACM Transactions on Networking*, 22(6):1938–1951, Dec 2014.

[MSY12] S. T. Maguluri, R. Srikant, and L. Ying. Stochastic models of load balancing and scheduling in cloud computing clusters. In *INFOCOM, 2012 Proceedings IEEE*, pp. 702–710, March 2012.

[NB10] A. Nedić and D. P. Bertsekas. The Effect of Deterministic Noise in Subgradient Methods. *Mathematical Programming*, 125(1):75–99, 2010.

[Ned02] A. Nedić. *Subgradient Methods for Convex Minimization*. PhD thesis, Massachusetts Institute of Technology, 2002.

[Nee06] M. J. Neely. Energy optimal control for time-varying wireless networks. *IEEE Transactions on Information Theory*, 52(7):2915–2934, July 2006.

[Nee10a] M. J. Neely. Stability and Capacity Regions or Discrete Time Queueing Networks. *ArXiv e-prints*, March 2010.

[Nee10b] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[Nee14] M. J. Neely. A Simple Convergence Time Analysis of Drift-Plus-Penalty for Stochastic Optimization and Convex Programs. *ArXiv e-prints*, December 2014.

[NML08] M. J. Neely, E. Modiano, and C. P. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, 16(2):396–409, April 2008.

[NMR03] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multibeam satellites with time-varying channels. *IEEE/ACM Transactions on Networking*, 11(1):138–152, Feb 2003.

[NMR05] M. J. Neely, E. Modiano, and C. E. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):89–103, 2005.

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[NO09a] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2016/07/09 2009.

[NO09b] A. Nedić and A. Ozdaglar. Subgradient Methods for Saddle-Point Problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009.

[NTS12] J. Ni, B. Tan, and R. Srikant. Q-csma: Queue-length-based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks. *IEEE/ACM Transactions on Networking*, 20(3):825–836, June 2012.

[NWZ13] J. Nair, A. Wierman, and B. Zwart. The fundamentals of heavy-tails: Properties, emergence, and identification. *SIGMETRICS Perform. Eval. Rev.*, 41(1):387–388, June 2013.

[NY78] A. Nemirovskii and D. Yudin. Cezare convergence of gradient method approximation of saddle points for convex-concave functions. *Doklady Akademii Nauk SSSR*, pp. 1056–1059, 1978.

[Pap80] M. Papageorgiou. A New Approach to Time-Of-Day Control Based on a Dynamic Freeway Traffic model. *Transportation Research Part B: Methodological*, 14(4):349–360, 1980.

[PB14] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[PC06] D. P. Palomar and M. Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, Aug 2006.

[Pol77] B. Polyak. Subgradient methods: A survey of soviet research. In *Nonsmooth optimization: Proceedings of the IIASA workshop March*, pp. 5–30, 1977.

[Pre95] A. Prekopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

[RNV09] S. S. Ram, A. Nedić, and V. V. Veeravalli. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717, 2009.

[Roc74] R. T. Rockafellar. *Conjugate duality and optimization*, volume 16. Siam, 1974.

[Roc84] R. T. Rockafellar. *Network flows and monotropic optimization*. Wiley, New York, NY, 1984.

[Roc93] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2):183–238, 1993.

[Roc97] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

[RV74] A. Roberts and D. Varberg. Another proof that convex functions are locally lipschitz. *The American Mathematical Monthly*, 81(9):1014–1016, 1974.

[RW98] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317. Springer: Grundlehren der Math. Wissenschaften., Berlin, 1998.

[S+58] M. Sion et al. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958.

[Sho85] N. Shor. *Minimization methods for nondifferentiable functions*. Translated from Russian by K.C. Kiwiel and A. Ruszczynski, Springer, Berlin, 1985.

[Sho12] N. Z. Shor. *Minimization methods for non-differentiable functions*, volume 3. Springer Science & Business Media, 2012.

[SJR16] A. Simonetto and H. Jamali-Rad. Primal recovery from consensus-based dual decomposition for distributed convex optimization. *Journal of Optimization Theory and Applications*, 168(1):172–197, 2016.

[SN11] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772–790, Aug 2011.

[Sto05] A. L. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[Sto06] A. L. Stolyar. Greedy primal-dual algorithm for dynamic resource allocation in complex networks. *Queueing Systems*, 54(3):203–220, 2006.

[SY13] R. Srikant and L. Ying. *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[TE92] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, 1992.

[Tse91] P. Tseng. On the rate of convergence of a partially asynchronous gradient projection algorithm. *SIAM Journal on Optimization*, 1(4):603–619, 1991.

[Tsi84] J. N. Tsitsiklis. *Problems in Decentralized Decision making and Computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[vdVBS09] P. van de Ven, S. Borst, and S. Shneer. Instability of maxweight scheduling algorithms. In *INFOCOM 2009, IEEE*, pp. 1701–1709, April 2009.

[vdVBY11] P. van de Ven, S. Borst, and L. Ying. Spatial inefficiency of maxweight scheduling. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, pp. 62–69, May 2011.

[VL16] V. Valls and D. J. Leith. Max-weight revisited: Sequences of nonconvex optimizations solving convex optimizations. *IEEE/ACM Transactions on Networking*, 24(5):2676–2689, October 2016.

[WSP07] X. Wu, R. Srikant, and J. R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Transactions on Mobile Computing*, 6(6):595–605, June 2007.

[WUW+12] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang. Distributed Traffic Signal Control for Maximum Network Throughput. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 588–595. IEEE, 2012.

[YJ10] B. Yang and M. Johansson. *Distributed Optimization and Games: A Tutorial Overview*, pp. 109–148. Springer London, London, 2010.

[YN15] H. Yu and M. J. Neely. On the Convergence Time of the Drift-Plus-Penalty Algorithm for Strongly Convex Programs. *ArXiv e-prints*, March 2015.

[Zin03] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. Technical report, School of Computer Science, Carnegie Mellon University, 2003.