1

Introduction

Network models are used to describe power grids, cellular telecommunications systems, large-scale manufacturing processes, computer systems, and even systems of elevators in large office buildings. Although the applications are diverse, there are many common goals:

(i) In any of these applications one is interested in controlling delay, inventory, and loss. The crudest issue is *stability*: do delays remain bounded, perhaps in the mean, for all time?

(ii) Estimating performance, or comparing the performance of one policy over another. Performance is of course context-dependent, but common metrics are average delay, loss probabilities, or backlog.

(iii) Prescriptive approaches to policy synthesis are required. A policy should have reasonable complexity; it should be flexible and robust. *Robustness* means that the policy will be effective even under significant modeling error. *Flexibility* requires that the system respond appropriately to changes in network topology, or other gross structural changes.

In this chapter we begin in Section 1.1 with a survey of a few network applications, and the issues to be explored within each application. This is far from comprehensive. In addition to the network examples described in the Preface, we could fill several books with applications to computer networks, road traffic, air traffic, or occupancy evolution in a large building.[1]

Although complexity of the physical system is both intimidating and unavoidable in typical networks, for the purposes of control design it is frequently possible to construct models of reduced complexity that lead to effective control solutions for the physical system of interest. These idealized models also serve to enhance intuition regarding network behavior.

Section 1.2 contains an outline of the modeling techniques used in this book for control design and performance evaluation. Section 1.3 reviews some of the mathematical prerequisites required to read the remainder of this book.

[1] *Egress* from a building is in fact a topic naturally addressed using the techniques described in Chapter 7. See the 1981 paper by Smith and Towsley [452], and the collection of papers [424].

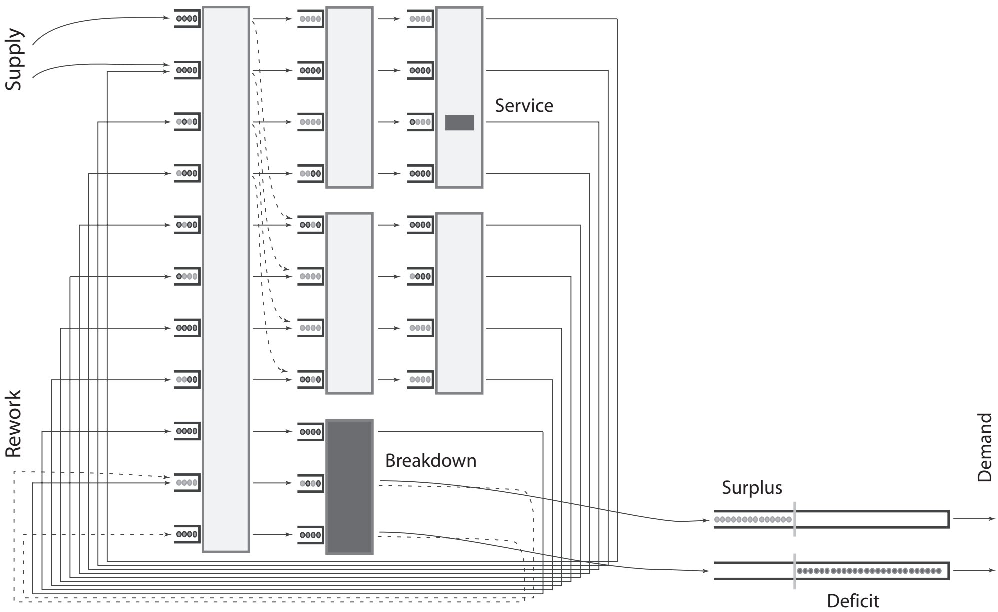

Figure 1.1. Control issues in a production system.

1.1 Networks in practice

1.1.1 Flexible manufacturing

Within the manufacturing domain, complexity is most evident in the manufacture of semiconductors.

A factory where semiconductors are produced is known as a wafer fabrication facility, or *wafer fab* [267, 410]. A schematic of a typical wafer fab is shown in Fig. 1.1. A large wafer fab will produce thousands of wafers each month, and a single wafer can hold thousands of individual semiconductor chips, depending on the size of the chips.

Control of a wafer fab or any other complex manufacturing facility involves many issues, including

(i) *Resource allocation:* Scheduling to minimize inventory, and satisfy constraints such as deadlines, finite buffers, and maximum processing rates. A key constraint in manufacturing applications is that one machine can only process one set of products at a time. This is significant in semiconductor manufacturing where one product (e.g., a wafer) may visit a single station repeatedly in the course of manufacture, and must complete with other products with similar requirements.

(ii) *Complexity management:* In the manufacture of semiconductors there may be hundreds of processing steps, and many different products. The control solution should have reasonable complexity in spite of the complexity of the system.

(iii) *Visualization of control solutions:* It is not enough for a solution to "spit out a sequence of numbers" representing service allocations at the stations in the manufacturing facility. Solutions should be tunable and provide some intuition to the user.

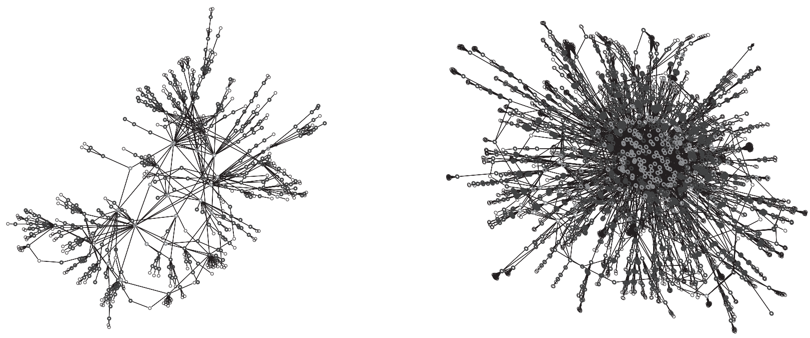

Figure 1.2. The Internet is one of the most complex man-made networks.

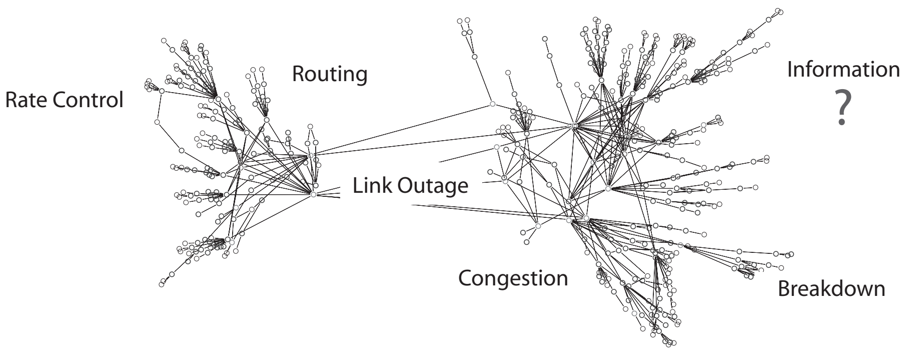

Figure 1.3. Control issues in the Internet.

(iv) *Recovery from crisis:* When machine failures occur, or preventative maintenance is called for, the control solution should be modified automatically.

(v) *Priorities:* The lifetime of a typical semiconductor wafer in a wafer fab may be more than 1 month. These typical wafers may sometimes compete with special *hot lots* that are given high priority due to customer demand or testing.

The International Semiconductor Roadmap for Semiconductors (ITRS) maintains a website describing the current challenges facing the semiconductor industry [279].

Scheduling policies are developed with each of these goals in mind in Chapters 4–7. Sometimes we are so fortunate that we can formulate policies that are nearly *optimal* when the network is highly loaded (see Chapter 9). Methods for approximating performance indicators such as mean delay are developed in Chapter 8.

1.1.2 The Internet

Figure 1.2 shows two subsets of the global Internet. Even a network representing a small Internet service provider (ISP) can show fantastic complexity.

As illustrated in Fig. 1.3, many issues arising in control of the Internet or more general communication networks are similar to those seen in production systems. In particular, decision making involves scheduling and routing of packets from node to node across a network consisting of links and buffers. Key differences are:

(i) Individual nodes do not have access to global information regarding buffer levels and congestion throughout the network. Routing or scheduling decisions must then be determined using only that information which can be made available.
(ii) Design is thus constrained by limited information. It is also constrained by protocols such as TCP/IP that are an inherent component of the existing Internet.
(iii) The future Internet will carry voice, audio, and data traffic. How can network resources be distributed fairly to a heterogenous customer population?

Burstiness and periodicity have been observed in Internet communications traffic [274, 504]. Part of the reason for these difficulties lies in the complex dynamics resulting from a large number of interconnected computers that are controlled based on limited local information.

In the future it will be possible to obtain much greater relevant information at each node in the network through *explicit congestion notification algorithms* [274, 182]. The system designer must devise algorithms to make use of this global information regarding varying congestion levels and network topology.

1.1.3 Wireless networks

It is evident today that wireless networks are only beginning to impact communications and computer networking. In a wireless network there are scheduling and routing decisions that are nearly identical to those faced in management of the Internet. The *resources* in a multiple-access wireless network include transmission power and bandwidth, as well as multiple paths between users and stations.

Wireless networks are subject to significant variability due to fading and path losses. Consequently, maximal transmission rates can be difficult to quantify, especially in a multiuser setting.

One significant difference between manufacturing and communication applications is that achievable transmission rates in a communication system depend upon the specific coding scheme employed. High transmission rates require long block-lengths for coding, which corresponds to long delays.

A second difference is that errors resulting from mutual interference from different users need not result in disaster, as would be the case in, say, transportation. Errors arising through collisions can be repaired through the miracle of coding, up to a point. These features make it difficult to quantify the capacity region in a communication networks, and wireless networks in particular.

1.1.4 Power distribution

Shown in Fig. 1.4 is a map of the California transmission network, which is of course embedded within the highly complex North American power grid.

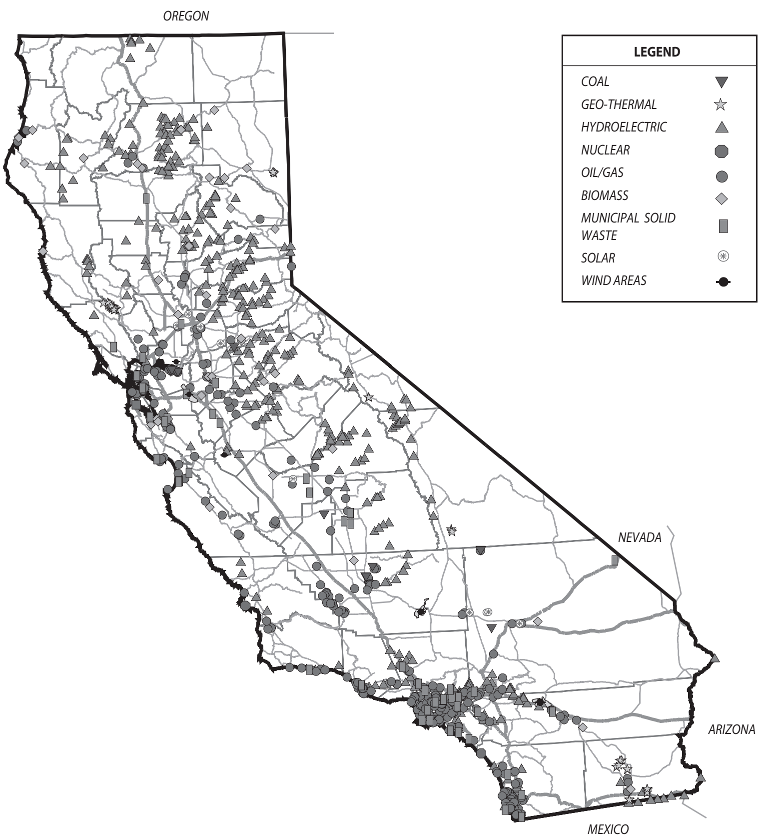

Figure 1.4. California power grid.

Regulation of power networks is further complicated by deregulation. Private power generators now provide a significant portion of electricity in the United States, whose owners seek to extract the maximal profit from the utilities who serve as their clients. However, the transmission network remains regulated by independent system operators (ISOs) who attempt to distribute transmission access fairly, and maintain system reliability.

Among the stated goals of deregulation are increased innovation, efficiency of power procurement, and reliability of power delivery. The results are often disappointing:

(i) During the period of 2000–2001, utilities in California saw historic price fluctuations and rolling blackouts. Suspicion of price manipulation was confirmed following the release of phone conversations in which ENRON employees discuss shutting down power plants in order to drive up prices [450, 91].

(ii) Reliability of the power grid is also dependent on the reliability of the electric transmission network. We are reminded of its importance by the major blackout of August 2003 that swept the north-eastern United States and parts of Canada [179, 168]. Similarly, wildfires in California in 2001 resulted in damaged

Cambridge University Press
978-0-521-88441-9 - Control Techniques for Complex Networks
Sean Meyn
Excerpt
More information

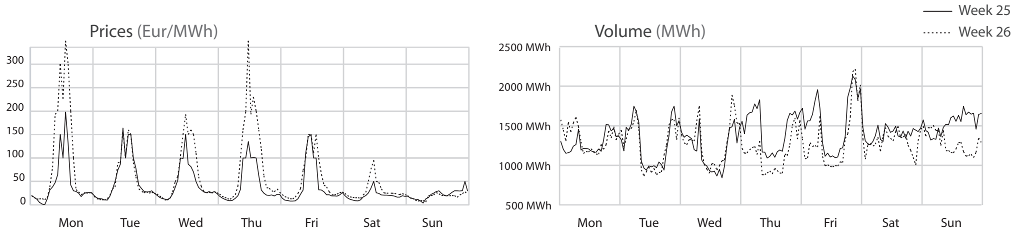

Figure 1.5. Market prices and power demand in continental Europe during the 25th and 26th weeks of 2003 (taken from the APX website [15]). Demand is periodic and shows high variability.

transmission lines that subsequently drove up power prices. Private conversations between ENRON employees reveal that they predicted these natural events would lead to increased profits [91].

(iii) A contributing factor to high power prices in California was the unusually hot and dry summer in 2000 [87]. This led to higher demand for power, and lower hydropower reserves. In a decentralized setting it is difficult to ensure that alternate sources of power reserves will be made available in the face of unexpected environmental conditions.

Even under average conditions, price and demand for power are periodic, and both exhibit significant variability. This is illustrated in Fig. 1.5 where demand and prices are plotted based on data collected in continental Europe during 2 weeks in 2003 [15]. The high volatility shown in these plots is *typical* behavior that has persisted for many years.

A power grid differs from many other network systems in that capacity must meet demand at every instant of time. If not, the transmission system may become unstable and collapse, with severe economic consequences to follow. For instance, according to the U.S. Department of Energy, the overall cost of the blackout of August 2003 was over 4 billion dollars [179, 168]. To ensure reliable operation it is necessary to schedule power generation capacity beyond the expected demand, called power reserves. Hence operation of the power grid is based on algorithms for forecasting demand, along with rules to determine appropriate power reserves.

The operational aspects of scheduling generation capacity in most power markets can be delineated into two general stages.

STAGE 1 The hour-by-hour demand for power can be predicted reasonably accurately over the upcoming 24 hour period. The high-capacity generators are scheduled in the first stage, on a day-ahead basis, based on these predictions. Advance planning is necessary since the high-capacity generators require time to ramp up or ramp down power production.

STAGE 2 The predicted demand is inevitably subject to error. To ensure system reliability, smaller generators are called upon in the second stage to provide a margin of *excess generation capacity*. These generators can ramp up or ramp down power production on a relatively short time scale, and hence can be scheduled on an hour-ahead basis.

A typical transmission network such as the California network may have hundreds of nodes, so a detailed model is far too complex to provide any insight into planning or design. On the other hand, the deterministic *DC power flow model* that is favored in many recent economic studies ignores important dynamic issues such as limited ramp-up rates, delayed information, and variability. The DC model can be viewed as a fluid equilibrium model, of the form used to define network load (see e.g. Chapter 4).

One of the goals of this book is to formulate compromise models that are simple enough for control design, and for performance approximation to compare control solutions.

1.2 Mathematical models

In each of these applications one is faced with a control problem: *what is the best way to sequence processing steps, or routing and scheduling decisions to obtain the best performance?*

An essential aspect of control theory is its flexible approach to modeling. For the purposes of design one typically considers a finite-dimensional, linear, deterministic system, even if the physical system is obviously nonlinear, with significant uncertainty with respect to modeling and disturbances. The idea is that the control system should be robust to uncertainty, so one should consider the simplest model that captures essential features of the system to be controlled.

1.2.1 A range of probabilistic models

The networks envisioned here consist of a finite set of stations, each containing a finite set of buffers. A *customer* residing at one of the buffers may represent a wafer, a packet, or a unit of power reserve. One or more *servers* process customers at a given station, after which a customer either leaves the network, or visits another station. Customers arrive from outside the network to various buffers in the network. The interarrival and service times all exhibit some degree of irregularity.

Consider a network with ℓ buffers, and ℓ_u different *activities* that may include scheduling, routing, or release of raw material into the system. Some of these buffers may be *virtual*. For example, in a manufacturing model, they may correspond to backlog or excess inventory. In a power distribution system, a buffer level is interpreted as the difference between the capacity and demand for power.

A general stochastic model can be described as follows: the vector-valued queue-length process $\boldsymbol{Q}$ evolves on $\mathbb{R}_+^\ell$, and the vector-valued cumulative allocation process $\boldsymbol{Z}$ evolves on $\mathbb{R}_+^{\ell_u}$. The ith component of $Z(t)$, denoted $Z_i(t)$, is equal to the cumulative time that the activity i has run up to time t. The evolution of the queue-length process is described by the vector equation

$$Q(t) = x + B(Z(t)) + A(t), \qquad t \geq 0, \quad Q(0) = x, \tag{1.1}$$

where the process $\boldsymbol{A}$ may denote a combination of exogenous arrivals to the network, and exogenous demands for materials *from* the network. The function $B(\,\cdot\,)$ represents the effects of (possibly random) routing and service rates.

The cumulative allocation process and queue-length process are subject to several hard constraints:

(i) The queue-length process is subject to the state space constraint

$$Q(t) \in \mathsf{X}, \qquad t \geq 0, \tag{1.2}$$

where $\mathsf{X} \subset \mathbb{R}_+^\ell$ is used to model finite buffers.

(ii) The control rates are subject to linear constraints

$$C(Z(t_1) - Z(t_0)) \leq \mathbf{1}(t_1 - t_0), \quad Z(t_1) - Z(t_0) \geq \mathbf{0}, \qquad 0 \leq t_0 \leq t_1\,, \tag{1.3}$$

where the *constituency matrix* C is an $\ell_m \times \ell_u$ matrix. The rows of C correspond to *resources* $r = 1, \ldots, \ell_m$, and the constraint (1.3) expresses the assumption that resources are shared among activities, and they are limited.

Stochastic models such as (1.1) have been by far the most popular in queueing theory. An idealization is the *linear fluid model*, described by the purely deterministic equation

$$q(t; x) = x + Bz(t) + \alpha t, \qquad t \geq 0,\ x \in \mathbb{R}_+^\ell, \tag{1.4}$$

where the state $\boldsymbol{q}$ evolves in the state space $\mathsf{X} \subset \mathbb{R}_+^\ell$, and the (cumulative) allocation process $\boldsymbol{z}$ evolves in $\mathbb{R}_+^{\ell_u}$. We again assume that $z(0) = \mathbf{0}$, and for each $0 \leq t_0 \leq t_1 < \infty$,

$$C[z(t_1) - z(t_0)] \leq (t_1 - t_0)\mathbf{1}, \quad \text{and} \quad z(t_1) - z(t_0) \geq \mathbf{0}. \tag{1.5}$$

The fluid model can also be described by the differential equation

$$\tfrac{d^+}{dt} q = B\zeta + \alpha, \tag{1.6}$$

where $\zeta = \zeta(t)$ denotes the right derivative, $\zeta = \frac{d^+}{dt} z$.

Two different symbols are used to denote the state processes for the stochastic and fluid models since much of the development to follow is based on the relationship between the two models. In particular, the fluid model can be interpreted as the mean flow of the stochastic model (1.1) on writing

$$Q(t) = x + A(t) - B(Z(t)) = x - BZ(t) + \alpha t + N(t), \qquad t \geq 0, \tag{1.7}$$

where α and B are interpreted as average values of $(\boldsymbol{A}, \boldsymbol{B})$, and

$$N(t) := [A(t) - \alpha t] - [B(Z(t)) - BZ(t)].$$

Typical assumptions on the stochastic model (1.1) imply that the mean of the process $\{N(t)\}$ is bounded as a function of time, and its variance grows linearly with t. Under these conditions (1.1) can be loosely interpreted as a fluid model subject to the additive disturbance $\boldsymbol{N}$.

1.2.2 What is a good model?

It is impossible to construct a model that provides an entirely accurate picture of network behavior. Statistical models are almost always based on idealized assumptions, such as independent and identically distributed (i.i.d.) interarrival times, and it is often difficult to capture features such as machine breakdowns, disconnected links, scheduled repairs, or uncertainty in processing rates.

In the context of economic modeling, Milton Friedman writes ... *theory is to be judged by its predictive power for the class of phenomena which it is intended to "explain*." The choice of an appropriate network model is also determined by its intended use. For long-range prediction the linear fluid model has little value, and for prediction alone a detailed model may be entirely suitable. Conversely, a model that gives an accurate representation of network behavior is likely to be far too complex to be useful for control design. Fortunately, it is frequently possible to create policies that are insensitive to modeling error, so that a design based on a relatively naive model will be effective in practice.

The controlled differential equation (1.6) can be viewed as a *state space model*, as frequently used in control applications, with control ζ, state q, and state space X. It is instructive to consider how control is typically conceptualized for linear systems without state-space constraints. Typically, a deterministic "fluid" model similar to (1.6) is taken as a starting point. If a successful design is obtained, then refinements are constructed based on a more detailed model that includes prior knowledge regarding uncertainty and noise. Once these issues are understood, the next step is to consider response to major structural uncertainty, such as component failure.

If the control engineers at NASA had not understood this point we never would have made it to the moon! In virtually *every* application of control, from flight control to cruise control, design is based on a fluid model described by an ordinary differential equation. This design is then refined to account for variability and other unmodeled quantities.

Throughout much of this book we adopt this control-theoretic viewpoint. We find that understanding a simple network model leads to practical solutions to many network control problems.

(i) Stability of the model of interest, in the sense of ergodicity, is essentially equivalent to a finite draining time for a fluid model. These connections are explored in Chapter 10.

(ii) Optimality of one is closely related to optimality of the other, with appropriate notions of "cost" for either model. In particular, the value function for the fluid control problem approximates the relative value function (the solution to Poisson's equation) for the discrete model (see Chapters 8 and 9).

In the control of linear state space models, Poisson's equation is known as the *Lyapunov equation*, and the solution is known to be a quadratic function of the state process when the cost is quadratic (see e.g. [329]). Remarkably, the solution is completely independent of variability, and moreover it coincides with the value function for an associated "fluid model."

(iii) In the case of network models, the value function for the deterministic fluid model is known as the *fluid value function*. This is a piecewise quadratic function when the cost function is linear. The solution to Poisson's equation for a stochastic network does not coincide with the fluid value function in general, but the two functions are approximately equal for large state values. This motivates the development of algorithms to construct quadratic or piecewise quadratic *approximations* to Poisson's equation for stochastic networks to bound steady-state performance. Deterministic algorithms are described in Chapter 8.

Approximate solutions to Poisson's equation such as a carefully chosen quadratic function, or the fluid value function, are used to construct fast simulation algorithms to estimate performance in Chapter 11.

(iv) A convenient approach to the analysis of buffer overflow or any similar disaster is through the analysis of a fluid model (see Section 3.5).

Again, when it comes to control design (i.e., policy synthesis), a solution obtained from an idealized model (deterministic or probabilistic) must be refined to account for unmodeled behavior. This refinement step for networks is developed over Chapters 4–11.

1.3 What do you need to know to read this book?

This book makes use of several different sets of tools from probability theory, control theory, and optimization.

1.3.1 Linear programs

In the theory of linear programming the standard *primal problem* is defined as the optimization problem

$$\begin{array}{lllll} \textbf{max} & c^{\mathsf T}x & & & \\ \textbf{s.t.} & \sum_j a_{ij}x_j & \le & b_i, & \text{for } i=1,\dots,m; \\ & x_j & \ge & 0, & \text{for } j=1,\dots,n. \end{array} \tag{1.8}$$

Its dual is the linear program

$$\begin{array}{lllll} \textbf{min} & b^{\mathsf T}w & & & \\ \textbf{s.t.} & \sum_j a_{ji}w_j & \ge & c_i, & \text{for } i=1,\dots,n; \\ & w_j & \ge & 0, & \text{for } j=1,\dots,m. \end{array} \tag{1.9}$$

The primal is usually written in matrix notation, $\max c^{\mathsf T}x$ subject to $Ax \le b$, $x \ge \mathsf{0}$; and the dual as $\min b^{\mathsf T}w$ subject to $A^{\mathsf T}w \ge c$, $w \ge \mathsf{0}$.

Any linear programming problem can be placed in the standard form (1.8). For example, a minimization problem can be reformulated as a maximization problem by