Chapter 1

Introduction to Process Optimization

Most things can be improved, so engineers and scientists optimize. While designing systems and products requires a deep understanding of influences that achieve desirable performance, the need for an *efficient and systematic decision-making approach* drives the need for optimization strategies. This introductory chapter provides the motivation for this topic as well as a description of applications in chemical engineering. Optimization applications can be found in almost all areas of engineering. Typical problems in chemical engineering arise in process design, process control, model development, process identification, and real-time optimization. The chapter provides an overall description of optimization problem classes with a focus on problems with continuous variables. It then describes where these problems arise in chemical engineering, along with illustrative examples. This introduction sets the stage for the development of optimization methods in the subsequent chapters.

1.1 Scope of Optimization Problems

From a practical standpoint, we define the *optimization* task as follows: given a system or process, find the best solution to this process within constraints. This task requires the following elements:

- An *objective function* is needed that provides a scalar quantitative performance measure that needs to be minimized or maximized. This can be the system's cost, yield, profit, etc.
- A predictive model is required that describes the behavior of the system. For the optimization problem this translates into a set of equations and inequalities that we term *constraints*. These constraints comprise a feasible region that defines limits of performance for the system.
- *Variables* that appear in the predictive model must be adjusted to satisfy the constraints. This can usually be accomplished with multiple instances of variable values, leading to a feasible region that is determined by a subspace of these variables. In many engineering problems, this subspace can be characterized by a set of *decision variables* that can be interpreted as *degrees of freedom* in the process.

Optimization is a fundamental and frequently applied task for most engineering activities. However, in many cases, this task is done by trial and error (through case study). To avoid such tedious activities, we take a systematic approach to this task, which is as efficient as possible and also provides some guarantee that a better solution cannot be found.

The systematic determination of optimal solutions leads to a large family of methods and algorithms. Moreover, the literature for optimization is dynamic, with hundreds of papers published every month in dozens of journals. Moreover, research in optimization can be observed at a number of different levels that necessarily need to overlap but are often considered by separate communities:

- At the *mathematical programming*[1] level, research focuses on understanding fundamental properties of optimization problems and algorithms. Key issues include existence of solutions, convergence of algorithms, and related issues such as stability and convergence rates.

- The *scientific computing* level is strongly influenced by mathematical properties as well as the implementation of the optimization method for efficient and "practical" use. Here research questions include numerical stability, ill-conditioning of algorithmic steps, and computational complexity and performance.

- At the level of *operations research*, attention is focused on formulation of the optimization problem and development of solution strategies, often by using well-established solution methods. Many of the problems encountered at this level consider well-structured models with linear and discrete elements.

- At the *engineering* level, optimization strategies are applied to challenging, and often poorly defined, real-world problems. Knowledge of optimization at this level is engaged with the efficiency and reliability of applicable methods, analysis of the solution, and diagnosis and recovery from failure of the solution method.

From the above description of optimization research, it is clear that successful development of an optimization strategy within a given level requires a working knowledge of the preceding levels. For instance, while it is important at the mathematical programming level to develop the "right" optimization *algorithm*, at the engineering level it is even more important to solve the "right" optimization *problem formulation*. On the other hand, as engineers need to consider optimization tasks on a regular basis, a systematic approach with a fundamental knowledge of optimization formulations and algorithms is essential. It should be noted that this requires not only knowledge of existing software, which may have limited application to particularly difficult problems, but also knowledge of the underlying algorithmic principles that allow challenging applications to be addressed. In the next section we begin with a classification of mathematical programming problems. This is followed by examples of optimization problems in chemical engineering that will be addressed in this text. Finally, a simple example is presented to motivate the development of optimization methods in subsequent chapters.

[1] The term *mathematical programming* was coined in the 1940s and is somewhat unrelated to computer programming; it originally referred to the more general concept of optimization in the sense of optimal planning.

1.2 Classification of Optimization Problems

Optimization is a key enabling tool for decision making in chemical engineering. It has evolved from a methodology of academic interest into a technology that continues to significant impact in engineering research and practice. Optimization algorithms form the core tools for (a) experimental design, parameter estimation, model development, and statistical analysis; (b) process synthesis, analysis, design, and retrofit; (c) model predictive control and real-time optimization; and (d) planning, scheduling, and the integration of process operations into the supply chain for manufacturing and distribution.

As shown in Figure 1.1, optimization problems that arise in chemical engineering can be classified in terms of continuous and discrete variables. When represented in algebraic form, the formulation of discrete/continuous optimization problems can be written as *mixed integer optimization problems*. The most general of these is the mixed integer nonlinear program (MINLP) of the form

$$\begin{aligned} \min_{x,y} \quad & f(x,y) \\ \text{s.t.} \quad & h(x,y) = 0, \\ & g(x,y) \leq 0, \\ & x \in \mathbb{R}^n, \quad y \in \{0,1\}^t, \end{aligned} \tag{1.1}$$

where $f(x,y)$ is the objective function (e.g., cost, energy consumption, etc.), $h(x,y) = 0$ are the equations that describe the performance of the system (e.g., material balances, production rates), and the inequality constraints $g(x,y) \leq 0$ can define process specifications or constraints for feasible plans and schedules. Note that the operator max $f(x,y)$ is equivalent to min $-f(x,y)$. We define the real n-vector x to represent the continuous variables while the t-vector y represents the discrete variables, which, without loss of generality, are often restricted to take $0/1$ values to define logical or discrete decisions, such as assignment of equipment and sequencing of tasks. (These variables can also be formulated to take on other integer values as well.) Problem (1.1) corresponds to an MINLP when any of the

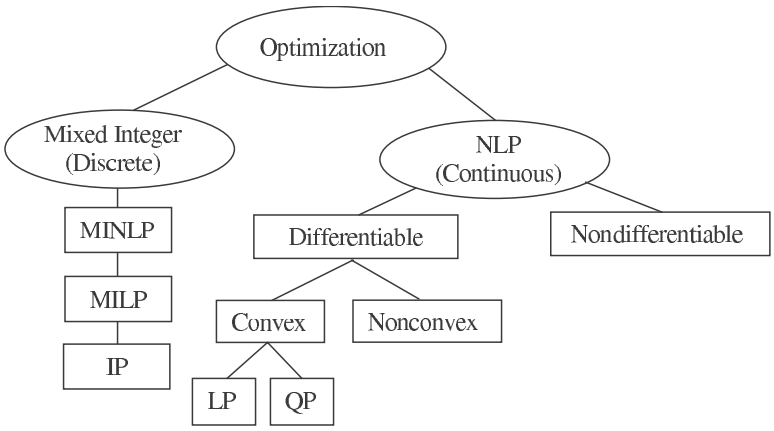

Figure 1.1. *Classes of optimization problems.*

functions involved are nonlinear. If the functions $f(x,y)$, $g(x,y)$, and $h(x,y)$ are linear (or vacuous), then (1.1) corresponds to a mixed integer linear program (MILP). Further, for MILPs, an important case occurs when all the variables are integer; this gives rise to an integer programming (IP) problem. IP problems can be further classified into a number of specific problems (e.g., assignment, traveling salesman, etc.), not shown in Figure 1.1.

If there are no 0/1 variables, then problem (1.1) reduces to the nonlinear program (1.2) given by

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{s.t.} & h(x)=0, \\ & g(x)\le 0. \end{array} \tag{1.2}$$

This general problem can be further classified. First, an important distinction is whether the problem is assumed to be differentiable or not. In this text, we will assume that the functions $f(x)$, $h(x)$, and $g(x)$ have continuous first and second derivatives. (In many cases, nonsmooth problems can be reformulated into a smooth form of (1.2).)

Second, a key characteristic of (1.2) is whether it is convex or not, i.e., whether it has a convex objective function, $f(x)$, and a convex feasible region. This can be defined as follows.

- A set $S\in\mathbb{R}^n$ is convex if and only if all points on the straight line connecting any two points in this set are also within this set. This can be stated as

$$x(\alpha)=\alpha x_1+(1-\alpha)x_2\in S \quad \text{for all } \alpha\in(0,1) \text{ and } x_1,x_2\in S. \tag{1.3}$$

- A function $\phi(x)$ is convex if its domain X is convex and

$$\alpha\phi(x_1)+(1-\alpha)\phi(x_2)\ge\phi(x(\alpha)) \tag{1.4}$$

  holds for all $\alpha\in(0,1)$ and all points $x_1,x_2\in X$. (Strict convexity requires that the inequality (1.4) be strict.)

- Convex feasible regions require $g(x)$ to be a convex function and $h(x)$ to be *linear.*

- A function $\phi(x)$ is (strictly) concave if the function $-\phi(x)$ is (strictly) convex.

If (1.2) is a convex problem, then any local solution (for which a better, feasible solution cannot be found in a neighborhood around this solution) is guaranteed to be a global solution to (1.2); i.e., no better solution exists. On the other hand, nonconvex problems may have multiple local solutions, i.e., feasible solutions that minimize the objective function only within some neighborhood about the solution.

Further specializations of problem (1.2) can be made if the constraint and objective functions satisfy certain properties, and specialized algorithms can be constructed for these cases. In particular, if the objective and constraint functions in (1.2) are linear, then the resulting linear program (LP) can be solved in a finite number of steps. Methods to solve LPs are widespread and well implemented. Currently, state-of-the-art LP solvers can handle millions of variables and constraints, and the application of further decomposition methods leads to the solution of problems that are two or three orders of magnitude larger than this. Quadratic programs (QPs) represent a slight modification of LPs through the addition of a quadratic term in the objective function. If the objective function is convex, then the

resulting convex QP can also be solved in a finite number of steps. While QP models are generally not as large or widely applied as LP models, a number of solution strategies have been created to solve large-scale QPs very efficiently. These problem classes are covered in Chapter 4.

Finally, we mention that while the nonlinear programming (NLP) problem (1.2) is given as a finite-dimensional representation, it may result from a possible large-scale discretization of differential equations and solution profiles that are distributed in time and space.

1.3 Optimization Applications in Chemical Engineering

Optimization has found widespread use in chemical engineering applications, especially in the engineering of process systems. Problems in this domain often have many alternative solutions with complex economic and performance interactions, so it is often not easy to identify the optimal solution through intuitive reasoning. Moreover, the economics of the system often indicate that finding the optimum solution translates into large savings, along with a large economic penalty for sticking to suboptimal solutions. Therefore, optimization has become a major technology that helps the chemical industry to remain competitive.

As summarized in Table 1.1, optimization problems are encountered in all facets of chemical engineering, from model and process development to process synthesis and design, and finally to process operations, control, scheduling, and planning.

Process development and modeling is concerned with transforming experimental data and fundamental chemical and physical relationships into a predictive process model. Incorporating the tasks of experimental design, parameter estimation, and discrimination of competing models, this step gives rise to NLP and MINLP problems.

At the next level, process synthesis and design incorporates these predictive models to devise a process that is both technically and economically feasible. The synthesis step is largely focused on establishing the structure of the process flowsheet (usually with simplified models) and systematic strategies have been developed for particular subsystems that involve reaction, separation, energy management, and waste reduction. On the other hand, the design step is concerned with establishing equipment parameters and nominal operating conditions for the flowsheet. As a result, synthesis problems can be addressed with a wide range of optimization formulations, while the design step requires more detailed models that give rise to NLP and MINLP problems.

In the operation of chemical processes, there is widespread interest in improving the scheduling and planning of these operations. These tasks are usually formulated as LP and MILP problems which are less detailed, as most operations are described through time requirements and activities. On the other hand, the development of large-scale NLP tools has led to the application of real-time optimization which interacts with the process and responds to changes in the production schedule and from other inputs. The results of the real-time optimization also feed operating conditions or *setpoints* to the control system. The distinction between these two levels is that real-time optimization describes steady state behavior, while the control system responds essentially to the process dynamics. The control task addresses maintenance of optimal production levels and rejection of disturbances. For multivariable, constrained systems, model predictive control (MPC) is now accepted as a widely used optimization-based strategy. MPC models allow various levels of detail to handle linear and nonlinear dynamics as well as discontinuities that occur in hybrid systems.

Table 1.1. *Mathematical programming in process systems engineering.*

	LP	MILP	QP	NLP	MINLP
Process Model Building				X	X
Process Design & Synthesis					
Heat Exchangers	X	X		X	X
Mass Exchangers	X	X		X	X
Separations		X		X	X
Reactors	X			X	X
Flowsheeting				X	X
Process Operations					
Scheduling	X	X			X
Supply Chain	X	X			X
Real-Time Optimization	X		X	X	
Process Control					
Model Predictive Control	X		X		
Nonlinear MPC			X	X	
Hybrid MPC		X			

Table 1.1 summarizes model types that have been formulated for process engineering applications. Design is dominated by NLP and MINLP models due to the need for the explicit handling of performance equations, although simpler targeting models for synthesis give rise to LP and MILP problems. Operations problems, in contrast, tend to be dominated by linear models, LPs and MILPs, for planning, scheduling, and supply chain problems. Nonlinear Programming, however, plays a crucial role at the level of real-time optimization. Finally, process control has traditionally relied on LP and NLP models, although MILPs are being increasingly used for hybrid systems. It is also worth noting that the applications listed in Table 1.1 have been facilitated not only by progress in optimization algorithms, but also by modeling environments such as GAMS [71] and AMPL [148].

This book focuses on the nonlinear programming problem (1.2) and explores methods that locate local solutions efficiently. While this approach might first appear as a restricted form of optimization, NLPs have broad applications, particularly for large-scale engineering models. Moreover, while the study of NLP algorithms is important on its own, these algorithms also form important components of strategies for MINLP problems and for finding the global optimum of nonconvex problems.

1.4 Nonlinear Programming Examples in Chemical Engineering

At the heart of any of the tasks in Table 1.1 lies a performance model of the process. Whether based on first principles or empirical relations, the model represents the behavior of reaction, separation, and mass, heat and momentum transfer in the process. The model equations for these phenomena typically consist of conservation laws (based on mass, heat,

and momentum), physical and chemical equilibrium among species and phases, and additional constitutive equations that describe the rates of chemical transformation or transport of mass and energy.

Chemical process models are often represented by a collection of individual unit models (the so-called unit operations) that usually correspond to major pieces of process equipment. Unit models are assembled within a process flowsheet that describes the interaction of equipment either for steady state or dynamic behavior. As a result, models can be described by algebraic or differential equations. For example, steady state process flowsheets are usually described by lumped parameter models described by algebraic equations. Similarly, dynamic process flowsheets are described by lumped parameter models described by differential-algebraic equations (DAEs). Models that deal with spatially distributed models are frequently considered at the unit level, with partial differential equations (PDEs) that model fluid flow, heat and mass transfer, and reactions. On the other hand, distributed models are usually considered too expensive to incorporate within an overall process model (although recent research [421, 243] is starting to address these issues).

Process models may also contain stochastic elements with uncertain variables. While these features are beyond the scope of this text, Chapter 6 considers a limited treatment of uncertainty through the formulation and efficient solution of multiple scenario optimization problems. These formulations link multiple instances of process models together with common variables. Similarly, models can also be linked to include multiple processes over multiple time scales. As a result, interactions between different operating levels (see Table 1.1) or among spatially distributed processes can be exploited through an optimization formulation.

To illustrate the formulation of NLPs from process models we introduce three examples from design, real-time optimization, and control. While solution strategies and results are deferred to later chapters, some detail is provided here to demonstrate both the characteristics of the resulting NLPs and some challenges in their solution.

1.4.1 Design of a Small Heat Exchanger Network

Consider the optimization of the small process network shown in Figure 1.2 with two process streams and three heat exchangers. Using temperatures defined by $T_{in} > T_{out}$ and $t_{out} > t_{in}$, the "hot" stream with a fixed flow rate F and heat capacity C_p needs to be cooled from T_{in} to T_{out}, while the "cold" stream with fixed flow rate f and heat capacity c_p needs to be heated from t_{in} to t_{out}. This is accomplished by two heat exchangers; the heater uses steam at temperature T_s and has a heat duty Q_h, while the cooler uses cold water at temperature T_w and has a heat duty Q_c. However, considerable energy can be saved by exchanging heat between the hot and cold streams through the third heat exchanger with heat duty Q_m and hot and cold exit temperatures, T_m and t_m, respectively.

The model for this system is given as follows:

- The energy balance for this system is given by

$$Q_c = FC_p(T_m - T_{out}), \tag{1.5}$$

$$Q_h = fc_p(t_{out} - t_m), \tag{1.6}$$

$$Q_m = fc_p(t_m - t_{in}) = FC_p(T_{in} - T_m). \tag{1.7}$$

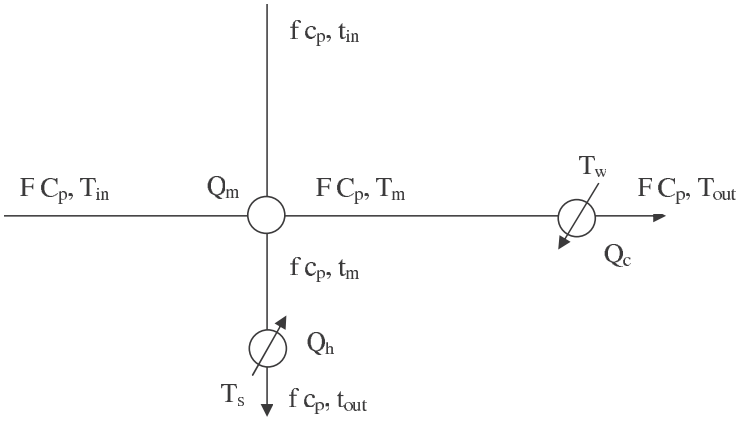

Figure 1.2. *Example of simple heat exchanger network.*

- Each heat exchanger also has a capital cost that is based on its area A_i, $i \in \{c,h,m\}$, for heat exchange. Here we consider a simple countercurrent, shell and tube heat exchanger with an overall heat transfer coefficient, U_i, $i \in \{c,h,m\}$. The resulting area equations are given by

$$Q_i = U_i A_i \Delta T_{lm}^i, \quad i \in \{c,h,m\}. \tag{1.8}$$

- The log-mean temperature difference ΔT_{lm}^i is given by

$$\Delta T_{lm}^i = \frac{\Delta T_a^i - \Delta T_b^i}{\ln(\Delta T_a^i / \Delta T_b^i)}, \quad i \in \{c,h,m\}, \tag{1.9}$$

  and

$$\begin{aligned} \Delta T_a^c &= T_m - T_w, \quad \Delta T_b^c = T_{out} - T_w, \\ \Delta T_a^h &= T_s - t_m, \quad \Delta T_b^h = T_s - t_{out}, \\ \Delta T_a^m &= T_{in} - t_m, \quad \Delta T_b^m = T_m - t_{in}. \end{aligned}$$

Our objective is to minimize the total cost of the system, i.e., the energy cost as well as the capital cost of the heat exchangers. This leads to the following NLP:

$$\min \quad \sum_{i \in \{c,h,m\}} (\hat{c}_i Q_i + \bar{c}_i A_i^\beta) \tag{1.10}$$

$$\text{s.t.} \quad (1.5)\text{–}(1.9), \tag{1.11}$$

$$Q_i \geq 0,\ \Delta T_a^i \geq \epsilon,\ \Delta T_b^i \geq \epsilon, \quad i \in \{c,h,m\}, \tag{1.12}$$

where the cost coefficients $\hat{c}_i$ and $\bar{c}_i$ reflect the energy and amortized capital prices, the exponent $\beta \in (0,1]$ reflects the economy of scale of the equipment, and a small constant $\epsilon > 0$ is selected to prevent the log-mean temperature difference from becoming

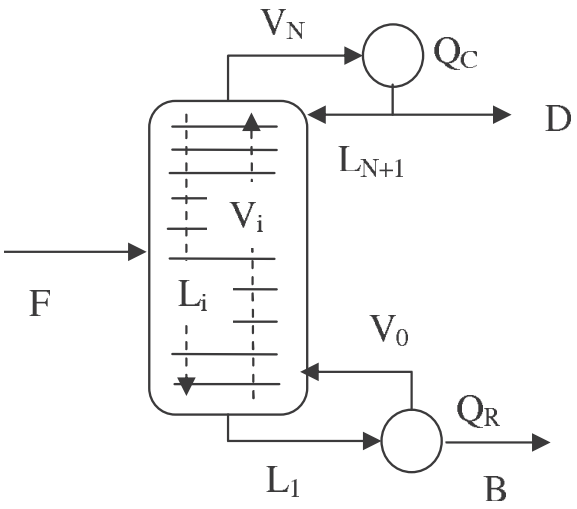

Figure 1.3. *Distillation column example.*

undefined. This example has one degree of freedom. For instance, if the heat duty Q_m is specified, then the hot and cold stream temperatures and all of the remaining quantities can be calculated.

1.4.2 Real-Time Optimization of a Distillation Column

Distillation is the most common means for separation of chemical components and lies at the heart of petroleum refining; it has no moving parts and scales easily and economically to all production levels. However, distillation is highly energy intensive and can consume 80%–90% of the total energy in a typical chemical or petrochemical process. As a result, optimization of distillation columns is essential for the profitability of these processes. Moreover, because distillation feeds, product demands and even ambient conditions change over time, the real-time optimization in response to these changes is also a key contributor to successful operation. Consider the distillation column shown in Figure 1.3 with N trays. As seen in the figure, liquid and vapor contact each other and approach equilibrium (i.e., boiling) on each tray. Moreover, the countercurrent flow of liquid and vapor provides an enrichment of the volatile (light) components in the top product and the remaining components in the bottom product. Two heat exchangers, the top condenser and the bottom reboiler, act as sources for the condensed liquid vapor and boiled-up vapor, respectively. The hydrocarbon feed contains chemical components given by the set

$$\mathcal{C} = \{\textit{propane, isobutane, n-butane, isopentane, n-pentane}\}.$$

The column is specified to recover most of the *n-butane* (the light key) in the top product and most of the *isopentane* (the heavy key) in the bottom product. We assume a total condenser and partial reboiler, and that the liquid and vapor phases are in equilibrium. A tray-by-tray

distillation column model is constructed as follows using the MESH (Mass-Equilibrium-Summation-Heat) equations:

Total Mass Balances

$$B + V_0 - L_1 = 0, \tag{1.13}$$

$$L_i + V_i - L_{i+1} - V_{i-1} = 0, \quad i \in [1, N], \ i \notin \mathcal{S}, \tag{1.14}$$

$$L_i + V_i - L_{i+1} - V_{i-1} - F = 0, \quad i \in \mathcal{S}, \tag{1.15}$$

$$L_{N+1} + D - V_N = 0. \tag{1.16}$$

Component Mass Balances

$$B x_{0,j} + V_0 y_{0,j} - L_1 x_{1,j} = 0, \quad j \in \mathcal{C}, \tag{1.17}$$

$$L_i x_{i,j} + V_i y_{i,j} - L_{i+1} x_{i+1,j} - V_{i-1,j} y_{i-1,j} = 0, \quad j \in \mathcal{C}, \quad i \in [1, N], \ i \notin \mathcal{S}, \tag{1.18}$$

$$L_i x_{i,j} + V_i y_{i,j} - L_{i+1} x_{i+1,j} - V_{i-1} y_{i-1,j} - F x_{F,j} = 0, \quad j \in \mathcal{C}, \ i \in \mathcal{S}, \tag{1.19}$$

$$(L_{N+1} + D) x_{N+1,j} - V_N y_{N,j} = 0, \quad j \in \mathcal{C}, \tag{1.20}$$

$$x_{N+1,j} - y_{N,j} = 0, \quad j \in \mathcal{C}. \tag{1.21}$$

Enthalpy Balances

$$B H_B + V_0 H_{V,0} - L_1 H_{L,1} - Q_R = 0, \tag{1.22}$$

$$L_i H_{L,i} + V_i H_{V,i} - L_{i+1} H_{L,i+1} - V_{i-1} H_{V,i-1} = 0, \quad i \in [1, N], \ i \notin \mathcal{S}, \tag{1.23}$$

$$L_i H_{L,i} + V_i H_{V,i} - L_{i+1} H_{L,i+1} - V_{i-1} H_{V,i-1} - F H_F = 0, \quad i \in \mathcal{S}, \tag{1.24}$$

$$V_N H_{V,N} - (L_{N+1} + D) H_{L,D} - Q_C = 0. \tag{1.25}$$

Summation, Enthalpy, and Equilibrium Relations

$$\sum_{j=1}^{m} y_{i,j} - \sum_{j=1}^{m} x_{i,j} = 0, \quad i = 0, \ldots, N+1, \tag{1.26}$$

$$y_{i,j} - K_{i,j}(T_i, P, x_i) x_{i,j} = 0, \quad j \in \mathcal{C}, \ i = 0, \ldots, N+1, \tag{1.27}$$

$$H_{L,i} = \varphi_L(x_i, T_i), H_{V,i} = \varphi_V(y_i, T_i), \quad i = 1, \ldots, N, \tag{1.28}$$

$$H_B = \varphi_L(x_0, T_0), H_F = \varphi_L(x_F, T_F), H_{N+1} = \varphi_L(x_{N+1}, T_{N+1}), \tag{1.29}$$

where

i	tray index numbered starting from reboiler ($= 1$)
$j \in \mathcal{C}$	components in the feed. The most volatile (lightest) is *propane*
P	pressure in the column
$\mathcal{F} \in [1, N]$	set of feed tray locations in column, numbered from the bottom
F	feed flow rate
L_i / V_i	flow rate of liquid/vapor leaving tray i
T_i	temperature of tray i
H_F	feed enthalpy
$H_{L,i} / H_{V,i}$	enthalpy of liquid/vapor leaving tray i
x_F	feed composition
$x_{i,j}$	mole fraction j in liquid leaving tray i
$y_{i,j}$	mole fraction j in vapor leaving tray i
$K_{i,j}$	nonlinear vapor/liquid equilibrium constant
φ_V / φ_L	nonlinear vapor/liquid enthalpy function
D / B	distillate/bottoms flow rate
Q_R / Q_C	heat load on reboiler/condenser
$lk / hk \in \mathcal{C}$	light/heavy key components that determine the separation.

The feed is a saturated liquid with component mole fractions specified in the order given above. The column is operated at a constant pressure, and we neglect pressure drop across the column. This problem has 2 degrees of freedom. For instance, if the flow rates for V_0 and L_{N+1} are specified, all of the other quantities can be calculated from (1.13)–(1.27). The objective is to minimize the reboiler heat duty which accounts for a major portion of operating costs, and we specify that the mole fraction of the light key must be 100 times smaller in the bottom than in the top product. The optimization problem is therefore given by

$$\min Q_R \tag{1.30a}$$
$$\text{s.t.} \quad (1.13)\text{–}(1.27) \tag{1.30b}$$
$$x_{bottom,lk} \leq 0.01 x_{top,lk}, \tag{1.30c}$$
$$L_i, V_i, T_i \geq 0, \quad i = 1, \ldots, N+1, \tag{1.30d}$$
$$D, Q_R, Q_C \geq 0, \tag{1.30e}$$
$$y_{i,j}, x_{i,j} \in [0,1], \quad j \in \mathcal{C}, \quad i = 1, \ldots, N+1. \tag{1.30f}$$

Distillation optimization is an important and challenging industrial problem and it also serves as a very useful testbed for nonlinear programming. This challenging application can be scaled up in three ways; through the number of trays to increase overall size, through the number of components to increase the size of the equations per tray, and through phase equilibrium relations, which also increase the nonlinearity and number of equations on each tray. An extended version of this problem will be considered in Chapter 11.

1.4.3 Model Predictive Control

Model predictive control (MPC) is a widely used optimization-based control strategy because it applies to general multivariable systems and allows constraints both on the input

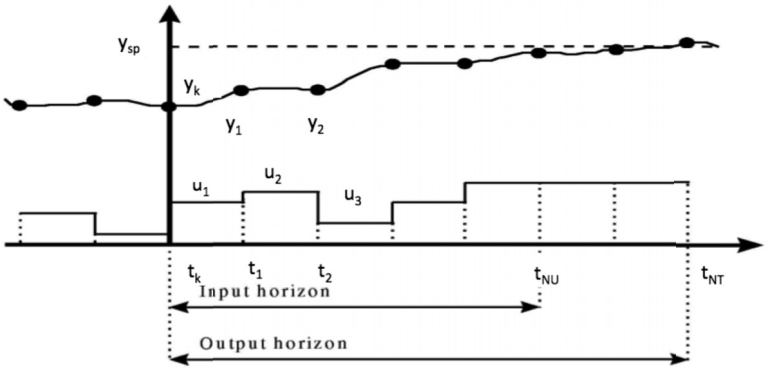

Figure 1.4. *Schematic for model predictive control.*

and output variables. In its most common form, a linear model is used to represent the dynamic process within separate time periods. A quadratic objective function is used to drive the output (or controlled) variables back to their desired values (or setpoints) and also to stabilize the control (or manipulated) variable profiles. As a result, a so-called quadratic programming problem is solved online. As seen in Figure 1.4 the control variables are allowed to vary in the first several time periods (H_u known as the input horizon) and the process is simulated for several more time periods (H_p known as the output horizon). The output horizon must be long enough so that the process will come back to steady state at the end of the simulation. After a horizon subproblem is solved, the control variables in the first time period are implemented on the process. At the next time period, the MPC model is updated with new measurements from the process and the next horizon subproblem is set up and solved again. The input horizon can be adjusted within the limits, $1 \leq H_u \leq H_p$. As long as the time sampling periods are short enough and the process does not stray too far from the setpoint (about which the system can be linearized), then a linear model usually leads to good controller performance [47]. On the other hand, for processes undergoing large changes including periods of start-up, shut-down, and change of steady states, the linear model for the process is insufficient, nonlinear models must be developed, and, as described in Chapter 9, a more challenging nonlinear program needs to be solved.

For most MPC applications, a linear time invariant (LTI) model is used to describe the process dynamics over time interval Δt; this is of the following form:

$$x^{k+1} = Ax^k + Bu^k, \tag{1.31}$$

where $x^k \in \mathbb{R}^{n_x}$ is the vector of state variables at $t = t_0 + k\Delta t$, $u^k \in \mathbb{R}^{n_u}$ is the vector of manipulated variables at $t = t_0 + k\Delta t$, $A \in \mathbb{R}^{n_x \times n_x}$ is the state space model matrix, $B \in \mathbb{R}^{n_x \times n_u}$ is the manipulated variable model matrix, $\Delta t \in \mathbb{R}$ is the sampling time, and $t_0 \in \mathbb{R}$ is the current time. A set of output (or controlled) variables is selected using the following linear transformation:

$$y^k = Cx^k + d^k, \tag{1.32}$$

where $y^k \in \mathbb{R}^{n_y}$ is the vector of output variables at $t = t_0 + k\Delta t$, $d^k \in \mathbb{R}^{n_y}$ is the vector of known disturbances at $t = t_0 + k\Delta t$, and $C \in \mathbb{R}^{n_y \times n_x}$ is the controlled variable mapping matrix.

The control strategy for this process is to find values for manipulated variables u^k, $k = 1, \ldots, H_u$, over an input horizon H_u such that (1.31)–(1.32) hold over the output horizon H_p for $k = 1, \ldots, H_p$ and $y^{H_p} \approx y^{\text{sp}}$ (i.e., endpoint constraint). To accomplish these goals, the following MPC QP is set up and solved:

$$\min_{u^k, x^k, y^k} \sum_{k=1}^{H_p} (y^k - y^{\text{sp}})^T Q_y (y^k - y^{\text{sp}}) + \sum_{k=1}^{H_u} (u^k - u^{k-1})^T Q_u (u^k - u^{k-1}) \tag{1.33a}$$

$$\text{s.t.} \quad x^k = Ax^{k-1} + Bu^{k-1} \quad \text{for } k = 1, \ldots, H_p, \tag{1.33b}$$

$$y^k = Cx^k + d^k \quad \text{for } k = 1, \ldots, H_p, \tag{1.33c}$$

$$u^k = u^{H_u} \quad \text{for } k = H_u + 1, \ldots, H_p \tag{1.33d}$$

$$-\hat{b} \le \hat{A}^T u^k \le \hat{b} \quad \text{for } k = 1, \ldots, H_p, \tag{1.33e}$$

$$u^L \le u_k \le u^U \quad \text{for } k = 1, \ldots, H_p \tag{1.33f}$$

$$-\Delta u^{\max} \le u^k - u^{k-1} \le +\Delta u^{\max} \quad \text{for } k = 1, \ldots, H_p, \tag{1.33g}$$

where $u^0 = u(t_0) \in \mathbb{R}^{n_u}$ is the initial value of manipulated variables, $y^{\text{sp}} \in \mathbb{R}^{n_y}$ is the vector of setpoints for output variables, $Q_y \in \mathbb{R}^{n_y \times n_y}$ is the diagonal penalty matrix for output variables, $Q_u \in \mathbb{R}^{n_u \times n_u}$ is the diagonal penalty matrix for manipulated variables, $\hat{b} \in \mathbb{R}^{n_u}$ are bounds for state constraints, $\hat{A} \in \mathbb{R}^{n_u \times n_u}$ is the transformation matrix for state constraints, and $\Delta u^{\max} \in \mathbb{R}^{n_u}$ is the maximum change allowed for the actuators.

This particular objective function (1.33a) allows the resulting MPC controller to drive the outputs to their setpoints and the terms for u^k help to regularize the QP and dampen wild changes in the manipulated variables. The constraints (1.33b) and (1.33c) are the process model and controlled variable selection equations, respectively. The constraints (1.33b)–(1.33d) form the dynamic model and are common to every MPC application. In particular, the constraints (1.33d) fix the control variables after the end of the input horizon H_u. The remaining constraints (1.33e)–(1.33g) are specific to the process. The ability of MPC to handle multivariable dynamic models along with additional bound constraints distinguishes this popular controller over its alternatives.

1.5 A Motivating Application

The above NLP problems can be solved with the algorithms presented in later chapters. However, to motivate the solution of these optimization problems we consider a smaller example shown in Figure 1.5. The cylindrical vessel has a fixed volume (V) given by $V = (\pi/4)D^2L$, and we would like to determine the length (L) and diameter (D) that minimize the cost of tank by solving the nonlinear program:

$$\min_{L,D} \; f \equiv c_s \pi D L + c_t (\pi/2) D^2 \tag{1.34a}$$

$$\text{s.t. } V = (\pi/4)D^2 L, \tag{1.34b}$$

$$D \ge 0, \quad L \ge 0. \tag{1.34c}$$

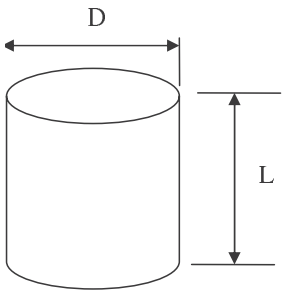

Figure 1.5. *Cylindrical tank optimization.*

Note that the cost of the tank is related to its surface area (i.e., the amount of material required) and c_s is the per unit area cost of the tank's side, while c_t is the per unit area cost of the tank's top and bottom.

This problem can be simplified by neglecting the bound constraints and eliminating L from the problem, leading to the unconstrained problem:

$$\min \; f \equiv 4c_s V/D + c_t(\pi/2)D^2. \tag{1.35}$$

Differentiating this objective with respect to D and setting the derivative to zero gives

$$\frac{df}{dD} = -4c_s V/D^2 + c_t \pi D = 0, \tag{1.36}$$

yielding $D = (\frac{4c_s V}{\pi c_t})^{1/3}$, $L = (\frac{4V}{\pi})^{1/3}(\frac{c_t}{c_s})^{2/3}$ and the aspect ratio $L/D = c_t/c_s$.

The optimal result helps to explain why soft drink cans are long and thin while storage tanks are short and fat. Moreover, we see that the solution of this problem is easily derived from simple differential calculus. Nevertheless, it is easy to see that the problem can be generalized so that an analytical solution may be much more difficult:

- The inequality constraints are neglected, and because L and D are both positive, they were satisfied (and inactive) at the solution. If this were not the case, more complicated optimality conditions would be applied, as derived in Chapter 4.
- The equality constraint that related L and D allowed an easy elimination of one of the variables. For nonlinear equations this is often not possible and an implicit elimination is needed, as described in Chapters 5 and 6.
- The variable D could be solved directly from the nonlinear equation (1.36) to yield a closed form solution. In many more applications, this is not possible and an indirect solution procedure must be required, as described in Chapter 3.

Dealing with these issues, especially as they apply to much larger problems, requires the numerical algorithms developed in this text. In particular, the last aspect in characterizing and finding solutions to unconstrained optimization problems is described in the next two chapters.

1.6 Summary and Notes for Further Reading

This chapter introduces the scope of optimization problems along with its conceptual components. Optimization covers a broad area, with a wide collection of problem classes. All of these can be found in chemical engineering applications. After a brief description of these areas, we focus on nonlinear programming problems and discuss specific application process examples of NLP. A small example is then covered in detail in order to motivate some of the issues in developing concepts and algorithms for NLP problems and to set the stage for the next chapters.

The overview in this chapter provides only a brief survey of process optimization applications. Modeling, analysis, design, and operations in chemical engineering are described in hundreds of textbooks and thousands of research sources. A reader unfamiliar with this literature is invited to sample *Perry's Chemical Engineers' Handbook*, known as the "chemical engineer's bible" [121], for a quick immersion into this area. Moreover, there is a growing list of textbooks that deal with optimization in chemical engineering, including the undergraduate texts [53, 122, 332]. All of these give a good overview of a broad set of optimization applications in chemical engineering.

In addition, optimization algorithms have been incorporated into a wide variety of optimization modeling platforms, including MATLAB®, GAMS [71], and AMPL [148], as well as widely used commercial chemical process simulators such as ASPEN, gPROMS, HySyS, Pro/II, and ROMeo. More information can be found on the websites of each vendor.

Finally, recent reviews [52, 175] provide a broader scope of the wealth of optimization problems in chemical engineering beyond NLP. These include MINLPs [53, 143, 295], global optimization algorithms and applications [144, 379], and stochastic programming [57, 216].

1.7 Exercises

1. Examine whether the following functions are convex or not:
   - $x^2 + ax + b$, $x \in \mathbb{R}$,
   - x^3, $x \in \mathbb{R}$,
   - x^4, $x \in \mathbb{R}$,
   - $\log(x)$, $x \in (0, 1]$.

2. Consider the minimization problem min $|x| + |y|$ s.t. $x^2 + y^2 = 1$. By introducing binary decision variables to handle the absolute value terms, reformulate this problem as an MILP of the form (1.1).

3. Show that the nonsmooth optimization problem min $\max(x, y)$ s.t. $x + y = 1, x, y \in \mathbb{R}$ has the same solution as

$$\begin{aligned} \min \quad & z \\ \text{s.t.} \quad & z \geq x, \\ & z \geq y, \\ & x + y = 1. \end{aligned}$$

4. Show that as $\Delta T_a^i \to \Delta T_b^i$, the log-mean temperature difference ΔT_{lm}^i given by (1.9) converges to the limit $\Delta T_{lm}^i = \Delta T_a^i$.

5. Modify the motivating example (1.34) and consider finding the minimum cost of a parallelepiped with a fixed volume. Show that the optimal solution must be a cube.

6. Consider the distillation column optimization (1.30) and introduce binary decision variables to formulate the optimal location of the feed tray as an MINLP of the form (1.1).