

1 Einleitung

Viele Probleme aus der Industrie und Wirtschaft sind Optimierungsprobleme, wie beispielsweise

- die möglichst billige Herstellung
- eines möglichst schnellen/sparsamen/robusten Autos.

Wir nennen solche Probleme im Folgenden „Anwendungsprobleme“.

Die Lösung des Anwendungsproblems lässt sich in zwei Arbeitsschritte gliedern, nämlich die Modellierung des Problems in mathematischer Form und die Lösung des mathematischen Problems.

1.1 Modellbildung, mathematische Formulierung

Eine Modellierung ist fast immer mit Idealisierungen verknüpft, das heißt das Anwendungsproblem wird in der Regel durch die mathematische Formulierung nur angenähert. Eine Lösung des mathematischen Problems ist daher entsprechend vorsichtig zu interpretieren. Oft haben die mathematischen Probleme keine Optimallösung oder keine eindeutige Optimallösung oder die Optimallösung kann nur näherungsweise ermittelt werden, was die Interpretation zusätzlich erschwert.

Die Modellbildung wird oft von den Anwendern wie Ingenieuren, Physikern, oder Unternehmern durchgeführt. Sie ist mindestens ebenso wichtig und schwierig wie die Lösung des mathematischen Problems, kann hier aber aufgrund der Vielfalt der einzelnen Anwendungen nicht näher beschrieben werden. Vielmehr soll in diesem Buch eine Einführung in die Theorie und Methoden der mathematischen Optimierung erfolgen und exemplarisch an einigen Anwendungsbeispielen im Bereich der stetigen und diskreten Optimierung die Anwendbarkeit der vorgestellten Verfahren aufgezeigt werden.

Typischerweise liefert die Modellbildung Systeme mit vielen Unbekannten, die wir in einem Vektor x passender Dimension zusammenfassen, der gewissen Nebenbedingungen in der Form von Gleichungen und Ungleichungen genügen muss. Durch Wahl von x lässt sich das System steuern. Das Verhalten des Systems wird durch eine reelle Zielfunktion f bewertet, die von x abhängt und die durch eine geeignete Wahl von x optimiert werden soll. Im Falle eines Minimierungsproblems führt dies zu dem mathematischen

Problem, den Funktionswert $f(x)$ unter allen x zu minimieren, die Nebenbedingungen der Form

$$\begin{aligned} f_i(x) &\leq 0 \quad \text{für } i \in I_1, \\ f_j(x) &= 0 \quad \text{für } j \in I_2, \\ x &\in \mathcal{B}. \end{aligned}$$

genügen. Wir schreiben für dieses Problem kurz

$$\begin{aligned} \inf \quad & f(x) \\ x : \quad & f_i(x) \leq 0 \quad \text{für } i \in I_1, \\ & f_j(x) = 0 \quad \text{für } j \in I_2, \\ & x \in \mathcal{B}. \end{aligned}$$

Hier sind I_1 und I_2 disjunkte Indexmengen, welche die Ungleichungs- und Gleichungsbedingungen „aufzählen“ und \mathcal{B} ist ein Bereich, auf dem f und alle f_i ($i \in I_1 \cup I_2$) als reelle Funktionen definiert sind. Darüber hinaus kann man mit Hilfe von \mathcal{B} weitere Bedingungen beschreiben, denen x genügen muss, die sich nicht als Konjunktion von Bedingungen in der Form einfacher Gleichungen oder Ungleichungen schreiben lassen (wie z.B. Ganzzahligkeitsbedingungen für gewisse Komponenten von x). Die Funktion f heißt *Zielfunktion*, die f_i und die Menge \mathcal{B} spezifizieren *Nebenbedingungen* (Restriktionen). Jeder Vektor x , der die Nebenbedingungen erfüllt, heißt *zulässige Lösung* des Problems. Diese Bezeichnung hat sich allgemein eingebürgert, auch wenn „zulässiger Punkt“ vielleicht passender wäre. Eine zulässige Lösung ist also in der Regel nicht die eigentlich gesuchte Lösung des Problems. Letztere werden mit *Optimallösung* bezeichnet. Optimallösungen sind also diejenigen zulässigen Lösungen, deren Wert $f(x)$ minimal ist.

1.2 Nichtlineare Programme

Der zweite Arbeitsschritt geht von der mathematischen Formulierung des Anwendungsproblems aus. Er befasst sich mit dessen Lösbarkeit und berechnet eine Optimallösung oder eine Näherung für eine Optimallösung. Wir beschränken uns in diesem Buch auf den Fall, dass der Vektor x der Unbekannten endlichdimensional ist, $x \in \mathbb{R}^n$, $\mathcal{B} \subset \mathbb{R}^n$, und dass nur endlich viele Nebenbedingungen zu beachten sind, d.h. auch die Indexmengen I_1 und I_2 sind endlich, etwa $I_1 = \{1, \dots, p\}$ und $I_2 = \{p+1, \dots, m\}$ mit $0 \leq p \leq m < \infty$. Wir erhalten dann das folgende Problem, für das sich die Bezeichnung *Nichtlineares „Programm“* (NLP) eingebürgert hat (passender wäre „nichtlineares Minimierungsproblem“):

$$(NLP) \quad \begin{aligned} \inf \quad & f(x) \\ x : \quad & f_i(x) \leq 0 \quad \text{für } i = 1, 2, \dots, p, \\ & f_j(x) = 0 \quad \text{für } j = p+1, p+2, \dots, m, \\ & x \in \mathcal{B}. \end{aligned}$$

Durch die Einschränkung auf $x \in \mathcal{B} \subset \mathbb{R}^n$ und auf endliche Mengen I_1, I_2 schließen wir interessante und sinnvolle Anwendungen aus, bei welchen z.B. x eine unbekannt Funktion ist (eine optimal zu wählende Steuerungsfunktion) oder bei welchen die Anzahl der Nebenbedingungen nicht endlich ist (semi-infinite Programme). Ebenso verzichten wir hier auf die Behandlung von Problemen aus der „multicriteria optimization“, wo mehrere verschiedene Zielfunktionen simultan zu berücksichtigen sind.

Das *NLP* in der obigen Form ist trotzdem noch sehr allgemein und es gibt keine Verfahren, Probleme dieser Allgemeinheit zufriedenstellend zu lösen. So macht selbst die Berechnung von

$$\inf_{x \in [0,1]} f(x), \quad \text{mit } f(x) := \sin \frac{1}{x + \epsilon} + \cos \frac{1}{(x + \epsilon)^2}$$

für jedes feste $\epsilon \in (0, 10^{-200})$ größte Schwierigkeiten (in diesem Beispiel ist $p = m = 0$, $\mathcal{B} = [0, 1] \subset \mathbb{R}^1$), obwohl f auf \mathcal{B} unendlich oft differenzierbar ist, nur von einer einzigen Unbekannten abhängt, und der „zulässige Bereich“ $[0, 1]$ kompakt ist.

Wir weisen weiter auf die Annahme hin, dass die Funktionen f und f_k , $1 \leq k \leq m$ auf der Menge

$$\mathcal{S} := \{x \in \mathcal{B} \mid f_i(x) \leq 0 \text{ für } 1 \leq i \leq p, f_j(x) = 0 \text{ für } p + 1 \leq j \leq m\}$$

der zulässigen Lösungen von (*NLP*) als reelle Funktionen definiert sein müssen. Als (*NLP*) formulierbar sind deshalb auch *mehrstufige* Optimierungsprobleme. Sie besitzen die Form

$$\inf \{f(x) \mid x \in \mathcal{S}\},$$

wobei $f(x)$ für $x \in \mathcal{S}$ selbst Lösung eines weiteren Optimierungsproblems

$$f(x) := \inf_{y \in \mathcal{S}_x} \phi(x, y)$$

ist, dessen zulässige Menge \mathcal{S}_x zudem von x abhängen kann. Bei diesen Problemen kann die Zielfunktion f sehr unangenehm sein; sie kann auch für differenzierbare ϕ nicht differenzierbar sein und selbst ihr Definitionsgebiet $\{x \mid \inf_{y \in \mathcal{S}_x} \phi(x, y) \in \mathbb{R}\}$ muss nicht a priori bekannt sein. Wir werden allerdings solche Optimierungsprobleme in diesem Buch nicht behandeln.

1.3 Einteilung von nichtlinearen Programmen

Für den Entwurf von Lösungsverfahren und die Beurteilung ihrer Leistungsfähigkeit ist es zweckmäßig, nichtlineare Programme in mehrere Klassen einzuteilen. Für jede Klasse lässt sich dann in gewissem Rahmen angeben, in wie weit man eine Lösung der entsprechenden Probleme mit heutigen Mitteln berechnen kann. Bei einem gegebenen *NLP* ist in der Regel eine gewisse Struktur erkennbar oder bekannt, die durch das Lösungsverfahren ausgenutzt wird. Eine grobe Einteilung der nichtlinearen Programme ist die folgende:

- 1) *Nichtrestringierte Minimierungsprobleme*, d.h. $p = m = 0$, $\mathcal{B} = \mathbb{R}^n$.
- 2) *Lineare Programme*, d.h. f und f_1, \dots, f_m sind affin und $\mathcal{B} = \mathbb{R}^n$.
- 3) *Konvexe Programme*, d.h. f und f_1, \dots, f_p sind konvexe Funktionen (s. Definition 2.5.2), f_{p+1}, \dots, f_m sind affin und $\mathcal{B} = \mathbb{R}^n$.
- 4) *Glatte, nichtlineare Programme*, d.h. f und f_1, \dots, f_m sind auf \mathbb{R}^n differenzierbar und $\mathcal{B} = \mathbb{R}^n$,
- 5) *Kombinatorische (diskrete) Probleme*. Diese lassen sich häufig als lineare Programme formulieren, bei denen $\mathcal{B} \neq \mathbb{R}^n$ ist und z.B. nur solche x enthält, für die gewisse Komponenten x_i ganzzahlig sind, oder noch spezieller, in $\{0, 1\}$ liegen.

Die obigen Klassen bilden nur eine unvollständige Grobeinteilung. Insbesondere ist es sinnvoll, die unter dem Oberbegriff der kombinatorischen Probleme zusammengefasste Klasse in weitere Unterklassen aufzuteilen, für welche jeweils spezielle Lösungsverfahren entwickelt worden sind.

Auch ist die angegebene Grobeinteilung nicht disjunkt, weil sie von der Formulierung des *NLP* abhängt; z.B. kann die Bedingung, dass \mathcal{B} nur solche x enthält, für die gewisse Komponenten x_i ganzzahlig sind, bzw. in $\{0, 1\}$ liegen, auch durch die Nebenbedingungen $f_{p+i}(x) := \sin \pi x_i = 0$ bzw. $f_{p+i}(x) := x_i^2 - x_i = 0$ ersetzt werden. Auf diese Weise lässt sich ein diskretes Problem sogar als ein glattes nichtlineares Programm schreiben. Ebenso lässt sich durch Einführung weiterer Variablen und Funktionen die Zahl der Ungleichungs- und Gleichungsrestriktionen ändern: So ist z.B. eine Ungleichung $f_i(x) \leq 0$ äquivalent zur Gleichung $f_i(x) + \bar{x}_i^2 = 0$, wobei \bar{x}_i eine neue Variable ist. Auch können mehrere Gleichungsrestriktionen, etwa $f_1(x) = 0, \dots, f_k(x) = 0$, zu einer Gleichungsrestriktion

$$f_1(x)^2 + \dots + f_k(x)^2 = 0$$

zusammengefasst werden. In den allermeisten Fällen „gewinnt“ man aber durch solche Umformungen nichts, weil das neue Problem nicht einfacher zu lösen ist. Die Umformung verdeutlicht aber die Feststellung, dass die Problemklassen nicht immer klar trennbar sind.

1.4 Ausblick

Wir beschäftigen uns in diesem Buch vorrangig mit der „stetigen Optimierung“ also in erster Linie mit den Klassen 1) bis 4). Dabei konzentrieren wir uns auf die Bestimmung lokaler Minima. Wichtige Anwendungen der Klasse 5), d.h. der kombinatorischen Optimierung, findet man z.B. in der Informatik und den Wirtschaftswissenschaften, während die Anwendungen für stetige Optimierungsprobleme oft aus den Ingenieurwissenschaften kommen. Auch methodisch unterscheiden sich die Lösungszugänge bei der stetigen und der kombinatorischen Optimierung. Viele Verfahren, die in der stetigen Optimierung eingesetzt werden, beruhen auf lokalen Approximationen der Zielfunktion und der Nebenbedingungen oder der Optimalitätsbedingungen mittels

Linearisierungen — wie z.B. das Newton-Verfahren aus der Schule. Verfahren, die in der kombinatorischen Optimierung zum Einsatz kommen, nutzen häufig geschickt gewählte „Probierstrategien“ sowie Ausschließungs- und Einschließargumente. Kombinatorische Probleme sind im allgemeinen schwieriger zu lösen und benötigen nicht selten eine Anzahl von Rechenschritten, die exponentiell mit der Anzahl der Unbekannten wächst. Wie wir sehen werden, lassen sich bei den Problemen der Klassen 2) und 3) wesentlich schnellere Algorithmen finden, welche im schlimmsten Fall eine Anzahl von Rechenschritten benötigen, die polynomial von der Anzahl der Unbekannten und der Anzahl der Nebenbedingungen abhängt.

Ein auf den ersten Anblick paradox wirkendes Phänomen ist dabei folgendes. Probleme der Klasse 2) oben (Lineare Programme) können sehr effizient gelöst werden. Sobald man das lineare Programm aber noch zusätzlich „vereinfacht“, indem man anstelle der reellen Zahlen ($\mathcal{B} = \mathbb{R}^n$) nur noch ganze Zahlen ($\mathcal{B} = \mathbb{Z}^n$) zulässt, ist das Problem (mit heutigen Mitteln) nicht mehr effizient lösbar. Ziel dieses Buches ist daher auch, den Leser für solche — und andere — „Vereinfachungen“ zu sensibilisieren.

Derzeit gibt es keine „guten“ Methoden, um allgemeine kombinatorische Probleme zu lösen. Hier bezeichnen wir ein Verfahren als gut, wenn es in „halbwegs vertretbarer“ Zeit stets eine Lösung des gestellten Problems finden kann. Allerdings könnte die sich abzeichnende Entwicklung von Quantencomputern dazu führen, dass auch kombinatorische Probleme mittelfristig effizient lösbar werden.

1.5 Zur Anwendung in der Praxis

Wir schließen diesen Abschnitt mit einer Bemerkung zu den Schwierigkeiten bei der Anwendung der Optimierung in der Praxis.

Die Implementierung der einzelnen Verfahren kann in diesem Buch nur in verkürzter Form vorgestellt werden. Dabei stecken gerade in der Implementierung noch sehr wesentliche Probleme, insbesondere bei der Ausnutzung der oftmals dünn besetzten Struktur der Eingabedaten (die bei der Verarbeitung der Daten im Laufe eines Verfahrens leicht verloren geht) und bei der Beherrschung von Rundungsfehlern. (Ein tragisches Beispiel für die Bedeutung von Rundungsfehlern ist z.B. der Einschlag einer amerikanischen Patriot-Rakete in einem amerikanischen Stützpunkt im Golfkrieg 1991, der auf Rundungsfehler zurückzuführen ist.) Wie wir später sehen werden, klafft zwischen Theorie und Praxis oft eine Lücke in dem Sinne, dass die Verfahren für die man die beste „worst-case“-Laufzeit beweisen kann, oft nicht die Verfahren sind, die in der Praxis die schnellsten Laufzeiten aufweisen. Insbesondere sind viele gebräuchliche Verfahren oft wesentlich besser als man es beweisen kann.

Typisch für Anwendungen in der Industrie sind folgende Punkte: Eine enge Zusammenarbeit mit anderen Disziplinen und die Einarbeitung in ein

spezielles Thema sind notwendig. Das Programm, das man als Mathematiker zunächst entwickelt, erfüllt häufig seinen Zweck nicht, weil sich die Problemstellung während der Programmentwicklung ändert, oder weil nicht alle Voraussetzungen bekannt waren. Eine Vielzahl von Änderungen am ersten Programmentwurf werden notwendig sein; das Programm muss daher von Anfang an sehr gut dokumentiert und möglichst modular und übersichtlich strukturiert sein. Das Programm wird mit anderen Programmen verknüpft werden, und die von auswärts bezogenen Programme sind nicht immer fehlerfrei. Der Nachweis, dass ein selbst entwickeltes Programm fehlerfrei ist, erfordert erhebliche zusätzliche Anstrengungen.

Wir werden solche Betrachtungen im Folgenden weitgehend außer Acht lassen und uns in erster Linie mit der Lösung der mathematischen Probleme befassen.