



Ralf Hollstein

Optimierungsmethoden

Einführung in die klassischen,
naturanalogen und neuronalen
Optimierungen



Springer Vieweg

Optimierungsmethoden

Ralf Hollstein

Optimierungsmethoden

Einführung in die klassischen, naturanalogen
und neuronalen Optimierungen

Ralf Hollstein
Linnich, Deutschland

ISBN 978-3-658-39854-5 ISBN 978-3-658-39855-2 (eBook)
<https://doi.org/10.1007/978-3-658-39855-2>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Der/die Herausgeber bzw. der/die Autor(en), exklusiv lizenziert an Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2023

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: David Imgrund

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Für Marita, Melanie und Leo

Vorwort

Optimierungen spielen in allen Lebensbereichen eine wichtige Rolle, sei es, um Kosten, Ressourcen, Risiken, Bearbeitungszeiten, Umweltbelastungen, Energieverbrauch, Reise- und Transportzeiten zu minimieren oder Gewinn, Motorleistung, Portfolio, Produktion, Umsatz und sportliche Leistungen zu maximieren. Die Optimierungsprobleme in der realen Welt sind vielfältig und unterscheiden sich in der Komplexität. In der Literatur gibt es eine unübersehbare Vielzahl von Optimierungsmethoden, davon wiederum Varianten und Untervarianten. Dieses Buch gibt einen Überblick zu den wichtigsten Optimierungsmethoden, wobei die Verfahren beispielhaft vorgestellt werden. Auf detaillierte Erläuterungen zur Theorie der in diesem Buch beschriebenen Verfahren wird verzichtet und auf die einschlägige Literatur verwiesen.

Zu vielen Optimierungsproblemen kann das Optimum bei größeren Eingabegrößen nicht in akzeptabler Zeit exakt bestimmt werden. Deshalb wendet man für komplexere Optimierungsprobleme sogenannte heuristische Verfahren an, die zwar nicht notwendigerweise die beste Lösung liefern, aber dafür in vertretbarer Zeit möglichst gute Lösungen finden. Beispielsweise ist in Kommunikationsnetzen beim dynamischen Routen von Datenpaketen nicht entscheidend, den kürzesten Weg zum Zielknoten zu bestimmen, sondern in möglichst kurzer Zeit einen möglichst kurzen Weg zu ermitteln. Bei vielen Optimierungsproblemen stoßen herkömmliche Großrechner an ihre Grenzen, wie zum Beispiel bei der Nutzung von Stromnetzen oder bei der optimalen Steuerung des Verkehrsflusses in einer Stadt. Hierbei muss der Verkehrsfluss dynamisch gesteuert werden, Staus müssen erkannt und Ersatzrouten bei ständig sich verändernden Optima gefunden werden. Dazu spielen effiziente Optimierungsalgorithmen eine wichtige Rolle.

Bei der Entwicklung von heuristischen Optimierungsverfahren werden gezielt Lösungen in der Biologie gesucht, analog wie in der Bionik, bei der zur Lösung von technischen Problemen Konstruktionen und Strukturen aus der Natur nachgeahmt werden. Dazu zählen beispielsweise der Klettverschluss nach dem Vorbild einer Klette, die Konstruktion von Flugzeugen nach dem Vorbild der Vögel, die selbstreinigenden Oberflächen nach dem Vorbild von Lotusblättern oder die Form

des japanischen Shinkansen-Schnellzuges nach dem Vorbild des Eisvogelschnabels zur Lärmreduzierung.

Bei der Entwicklung von naturinspirierten Optimierungsverfahren stehen die Verfahrensweisen der Natur im Vordergrund. Zu den bekanntesten Optimierungsmethoden zählen die evolutionären Algorithmen, die sich am Vorbild der natürlichen Evolution orientieren. Die Grundprinzipien der Evolution wie Mutation, Rekombination und Selektion lassen sich algorithmisch in Optimierungsverfahren umsetzen, die breite Anwendung in vielen Gebieten finden. Bei diesen naturanalogen Optimierungsverfahren werden Ausgangslösungen so lange verändert und kombiniert, bis eine dieser Lösungen zu den Anforderungen passt. Ebenfalls weitverbreitet sind Optimierungsmethoden, die das Verhalten von Schwärmen imitieren. Schwärme bilden eine Gruppe von vielen primitiven Lebewesen, die komplexe Fähigkeiten entwickeln können, was man als Schwarmintelligenz bezeichnet. Beobachtet man etwa Ameisen bei der Nahrungssuche, so kann man feststellen, dass die Ameisen von ihrem Nest zum Futterplatz meistens den kürzesten Weg finden. Die Optimierungsmethode, die diese Optimierungsstrategie nachahmt, konnte erfolgreich in vielen Bereichen angewendet werden, wie zum Beispiel bei der Tourenplanung. Es haben sich weitere schwarmbasierte Optimierungsmethoden etabliert, unter anderem Verfahren, die Optimierungsstrategien von Bienen und Vögeln imitieren. Ebenfalls lassen sich viele Probleme mithilfe von sogenannten künstlichen Immunsystemen lösen, die Konzepte natürlicher Immunsysteme bei der Bekämpfung von Antigenen wie Bakterien, Viren, Pilzen und Parasiten nachahmen.

Im ersten Teil werden verschiedene Optimierungsprobleme vorgestellt. Gegenstand des zweiten Teils sind klassische Methoden zur Lösung von Optimierungsproblemen. Im dritten Teil werden naturanaloge Optimierungsmethoden behandelt.

Gegenstand des letzten Teils des Buches sind Optimierungen von künstlichen neuronalen Netzen sowie Methoden des maschinellen Lernens zur Lösung von Optimierungsproblemen. Das künstliche neuronale Netz ist dem Aufbau des biologischen Gehirns nachempfunden. Es besteht aus Neuronen (Knoten), die über sogenannte Kanten mit anderen Neuronen verbunden sind und Informationen modifiziert an andere Neuronen weiterleiten.

Ein neues Forschungsgebiet sind neuronale kombinatorische Optimierungsverfahren, abgekürzt NCO. Dieser Begriff ist von Bello et al. 2015 eingeführt worden. Die NCO-Methoden stellen einen Strategie-Wechsel dar. Bei der herkömmlichen Programmierung einer Heuristik erhält der Computer Anweisungen zur Bestimmung einer brauchbaren Lösung des Optimierungsproblems. Dagegen ermittelt der Computer mit dem NCO-Verfahren selbstständig ohne menschliche Hilfe die Heuristik, wobei die Heuristik im Verborgenen bleibt. Dabei werden Verfahren aus dem Bereich des maschinellen Lernens angewendet. Dazu stehen vorprogrammierte Module aus Programmbibliotheken wie z. B. TensorFlow zur Verfügung. Der wesentliche Vorteil der NCO-Methoden besteht darin, dass diese Verfahren auf praktische Optimierungsprobleme anwendbar sind, zu denen keine guten Heuristiken existieren. Dieses Buch gibt eine Einführung in dieses neue vielversprechende Forschungsgebiet der KI-basierten, selbstlernenden Optimierungsalgorithmen.

Das Buch dient als Einstieg in die Optimierungsverfahren und wendet sich an Anwender auf den Gebieten der praktischen Optimierung sowie an Studierende der Informatik, Mathematik, Wirtschaftswissenschaften und Ingenieurwissenschaften.

Linnich
September 2022

Ralf Hollstein

Inhaltsverzeichnis

Teil I Optimierungsprobleme

1	Einführung	3
1.1	Optimierungsprobleme in der realen Welt	3
1.2	Grundbegriffe der Optimierung	5
1.2.1	Globales und lokales Optimum	5
1.2.2	Umwandlung eines Maximierungsproblems in ein Minimierungsproblem und umgekehrt	5
1.2.3	Notationen	6
1.2.4	Supremum und Infimum	6
1.2.5	Kontinuierliche, diskrete und kombinatorische Optimierung	7
2	Kontinuierliche Optimierungsprobleme	9
2.1	Graph einer Funktion	9
2.2	Optimierungen mit Nebenbedingungen	10
2.3	Anwendungsbeispiele	11
2.4	Unimodale und multimodale Funktionen	14
2.5	Konvexe und konkave Funktionen	14
2.6	Extrema konvexer und konkaver Funktionen	15
3	Kombinatorische Optimierungsprobleme	17
3.1	Einführende Beispiele	17
3.1.1	Problem des Handlungsreisenden	18
3.1.2	Rucksackproblem	19
3.2	Lösungspräsentationen	20
3.2.1	Kombinatorik	21
3.2.2	Graphen	23
3.3	Graphenprobleme	27
3.3.1	Knotenüberdeckungsproblem	28
3.3.2	Maximales Cliquenproblem	29
3.3.3	Stabilitätsproblem	30

3.3.4	Zusammenhang zwischen Stabilitäts-, Knotenüberdeckungs- und maximalem Cliquesproblem	32
3.3.5	Graphen-Färbungsproblem	33
3.3.6	Max-Cut-Problem	35
3.4	Vehicle-Routing-Probleme.	37
3.4.1	Einführendes Beispiel	37
3.4.2	Grundversion des VRP.	39
3.4.3	Varianten des VRP	40
3.5	Scheduling-Probleme.	41
3.5.1	Einführendes Beispiel	42
3.5.2	Grundversion des Scheduling-Problems	42
3.5.3	Klassifikation von Scheduling-Problemen.	44
3.5.4	Beispiele von Scheduling-Problemen	44
3.6	Zuschnittprobleme und Packungsprobleme.	45
3.6.1	Dimension von Zuschnitt- und Packungsproblemen	45
3.6.2	Bin-Packing-Problem.	46
3.6.3	Eindimensionale C&P-Probleme	46
3.6.4	Zweidimensionale C&P-Probleme	47
3.6.5	Dreidimensionale C&P-Probleme	49
	Literatur.	50
4	Lineare Optimierungsprobleme	51
4.1	Einführungsbeispiel (Produktionsproblem).	51
4.2	Standardform eines linearen Optimierungsproblems.	52
4.3	Anwendungsbeispiel (Transportproblem)	53
4.4	Ganzzahlige lineare Optimierungsprobleme	54
5	Multikriterielle Optimierungsprobleme	57
5.1	Definitionen	57
5.2	Multikriterielles Optimierungsproblem in der Grundversion	58
5.3	Einführendes Beispiel	59
5.4	Pareto-Dominanz	60
 Teil II Klassische Optimierungsmethoden		
6	Analytische Methoden	65
6.1	Grundbegriffe der Analysis	66
6.1.1	Definitionen	66
6.2	Der Satz vom Maximum und Minimum	67
6.3	Ableitung einer Funktion.	68
6.3.1	Sekanten- und Tangentensteigung	68
6.3.2	Ableitung einer Funktion	69
6.3.3	Beispiele von Ableitungen und Ableitungsregeln	69

6.4	Extremwertbestimmung für Funktionen mit einer Variablen. . .	70
6.5	Extremwertbestimmung für Funktionen mit zwei Variablen. . .	72
6.5.1	Partielle Ableitungen	72
6.5.2	Notwendige Bedingung für ein Extremum	73
6.5.3	Hinreichende Bedingung für lokale Extrema.	76
6.6	Lagrange-Methode.	78
6.6.1	Gradient einer Funktion	78
6.6.2	Richtungsableitung.	79
6.6.3	Höhenlinie	80
6.6.4	Eigenschaften des Gradienten	81
6.6.5	Lagrange-Funktion.	81
6.6.6	Langrange-Methode.	83
6.6.7	Anwendungsbeispiel	83
6.6.8	Lagrange-Methode für Funktionen mit mehr als zwei Variablen	84
6.7	Gradientenverfahren.	84
6.7.1	Ablauf des Gradientenabstiegsverfahrens	85
6.7.2	Schrittweitenbestimmung.	86
6.7.3	Gradientenverfahren zur Bestimmung lokaler Maxima	87
7	Methoden zur Lösung kombinatorischer Optimierungsprobleme.	89
7.1	Enumerationsmethode	90
7.2	Branch-and-Bound-Verfahren	91
7.2.1	Die Methode des Branch-and-Bound-Verfahrens . . .	91
7.2.2	Branch-and-Bound-Verfahren für das TSP	92
7.3	Dynamische Programmierung	93
7.4	Greedy-Algorithmus	97
7.4.1	Greedy-Algorithmus für das Münzwechsel-Problem.	97
7.4.2	Greedy-Algorithmus für ein Rucksackproblem.	98
7.4.3	Dijkstra-Algorithmus	99
7.5	Einfache lokale Suche	102
7.5.1	Nachbarschaft einer Lösung.	102
7.5.2	Methode der einfachen lokalen Suche	103
7.6	Tabu-Suche.	106
7.6.1	Einführendes Beispiel (Rucksackproblem)	106
7.6.2	Grundversion der Tabu-Suche	108
7.6.3	Varianten der Tabu-Suche	109
	Literatur.	109
8	Lineare Optimierung	111
8.1	Graphische Methode	111
8.2	Simplexmethode	114
8.2.1	Standard-Maximierungsproblem	114

8.2.2	Schlupfvariable	115
8.2.3	Austauschverfahren	115
8.2.4	Strategie der Simplexmethode	117
9	Multikriterielle Optimierungsverfahren	121
9.1	Methode der gewichteten Summe	121
9.2	Die ϵ -Constraint-Methode	123
9.3	Pareto-Optimierung mithilfe naturanaloger Optimierungsverfahren	124
10	Komplexität und heuristische/metaheuristische Verfahren	125
10.1	Komplexität	125
10.1.1	O-Notation	126
10.1.2	Beispiel (Suche in einer sortierten Liste)	126
10.1.3	Komplexität von Problemen	127
10.1.4	Polynomielle Algorithmen	128
10.1.5	Entscheidungsprobleme	128
10.1.6	Komplexitätsklassen P und NP	129
10.1.7	Polynomielle Reduzierbarkeit	129
10.1.8	Komplexitätsklasse NP-vollständig	131
10.1.9	NP-schwere Optimierungsprobleme	131
10.2	Heuristisches Verfahren	132
10.3	Metaheuristische Verfahren	133
10.4	Exploitation und Exploration	133
10.5	No Free Lunch Theorem	134
	Literatur	134
Teil III Naturanaloge Optimierungen		
11	Physikbasierende Algorithmen	137
11.1	Simulated Annealing	137
11.1.1	Abkühlungsprozess beim Erhärten einer Metallschmelze	138
11.1.2	Simulated-Annealing-Algorithmus	139
11.2	Threshold-Accepting-Algorithmus	142
11.3	Sintflut-Algorithmus	144
11.4	Populationbasierende Algorithmen	145
	Literatur	146
12	Evolutionäre Algorithmen	147
12.1	Biologische Evolution	147
12.2	Genetische Algorithmen	149
12.2.1	Grundversion der genetischen Algorithmen	149
12.3	Varianten der genetischen Algorithmen	155
12.3.1	Varianten von Selektionsverfahren	155
12.3.2	Varianten von Rekombinationen	158
12.3.3	Varianten von Mutationen	159

12.3.4	Varianten der Ersetzung (Replacement)	160
12.4	Permutationscodierung	161
12.4.1	Rekombination für Permutationen	162
12.4.2	Mutationen für Permutationen	164
12.4.3	Beispiel: n-Damen-Problem.	165
12.5	Anwendungen genetischer Algorithmen	166
12.5.1	Optimierung kontinuierlicher Funktionen mit GA	166
12.5.2	Multikriterielle Optimierung mit GA.	170
12.6	Evolutionsstrategien.	173
12.6.1	Der Grundalgorithmus der Evolutionsstrategie	174
12.6.2	Initialisierung	175
12.6.3	Rekombination.	175
12.6.4	Mutationen	176
12.6.5	Selektion.	183
12.6.6	Abbruch	183
	Literatur.	184
13	Partikelschwarmalgorithmen	185
13.1	Schwarmverhalten	185
13.2	Boids	186
13.3	Grundversion des Partikelschwarmalgorithmus	187
13.3.1	Einführendes Beispiel	188
13.3.2	Grundversion des PSO-Algorithmus	188
13.3.3	Ablauf der Grundversion des PSO.	190
13.4	Varianten des PSO-Algorithmus	191
13.4.1	Maßnahmen beim Verlassen des Suchraumes	191
13.4.2	Nachbarschafts-Topologien	192
13.4.3	Trägheitsparameter.	194
13.4.4	Kontraktionsfaktor	195
13.4.5	Tabelle der wichtigsten Parameter des PSO-Verfahrens	196
13.5	Anwendungsbeispiel (Diagnostik).	196
13.6	Diskrete Partikelschwarmoptimierungen.	196
13.6.1	Binäres PSO-Verfahren	197
13.6.2	DPSO-Verfahren	199
13.7	Multikriterielle Partikelschwarmoptimierung	201
	Literatur.	207
14	Ameisenalgorithmen	209
14.1	Ameisen in der Natur.	209
14.2	Grundprinzip des Ameisenalgorithmus	210
14.3	Die Grundversion des Ameisenalgorithmus	213
14.4	Der Algorithmus AS.	214
14.5	Unterschied zwischen realen und künstlichen Ameisen	216
14.6	Varianten des Ameisenalgorithmus AS	217

14.6.1	Elitist AS (EAS)	217
14.6.2	Rank based AS (ASrank)	218
14.6.3	MAX-MIN Ant-System (MMAS)	218
14.6.4	Ant Colony System (ACS)	219
14.7	Parameterwerte für ACO-Algorithmen	220
14.8	Anwendungen	220
14.8.1	Rucksackproblem	220
14.8.2	Generalised Assignment Problem	221
14.8.3	Routing in Netzwerken	223
14.8.4	Shortest Common Supersequence Problem	225
	Literatur	230
15	Bienenalgorithmen	231
15.1	Bienen in der Natur	231
15.1.1	Bienenvolk	231
15.1.2	Futtersuche der Bienen	232
15.2	Der BA-Algorithmus	234
15.2.1	Die Grundversion des BA-Algorithmus	234
15.2.2	Analogie zwischen natürlichen und künstlichen Bienen	235
15.2.3	Lokale und globale Suche	236
15.2.4	Anwendungsbeispiel	237
15.3	Der ABC-Algorithmus	239
	Literatur	241
16	Fledermausalgorithmen	243
16.1	Fledermäuse in der Natur	243
16.2	Grundversion des BAT-Algorithmus	244
16.2.1	Parameter des BAT-Algorithmus	244
16.2.2	BAT-Algorithmus in der Grundversion	245
16.2.3	Analogie zwischen natürlichen und künstlichen Fledermäusen	247
16.2.4	BAT-Algorithmus für das Problem des Handlungsreisenden	248
16.3	Binärer BAT-Algorithmus	250
16.4	Multikriterieller BAT-Algorithmus	251
	Literatur	252
17	Künstliche Immunsysteme	253
17.1	Natürliches Immunsystem	253
17.1.1	Angeborenes und erworbenes Immunsystem	254
17.1.2	Affinität	255
17.2	Positive und negative Selektion	256
17.2.1	Thymusmodell	257
17.2.2	Künstliche positive und negative Selektion	257

17.3	Klonaler Selektions-Algorithmus	260
17.3.1	Klonale Selektionstheorie	260
17.3.2	CLONALG für Mustererkennung	262
17.3.3	CLONALG für Optimierung	266
17.4	Immuner-Netzwerk-Algorithmus.	270
17.4.1	Die Jernesche Netzwerktheorie	270
17.4.2	Der aiNet-Algorithmus	271
17.4.3	Anwendung (Clusteranalyse).	273
17.4.4	Der opt-aiNet -Algorithmus.	274
	Literatur.	277
18	Übersicht: Naturanaloge Optimierungen	279
18.1	Naturanaloge Optimierungen in diesem Buch.	279
18.2	Übersicht weiterer naturanaloger Optimierungsmethoden	280
	Literatur.	281

Teil IV Neuronale kombinatorische Optimierung

19	Neuronale Netze	285
19.1	Natürliches neuronales Netz	285
19.2	Das künstliche Neuron.	286
19.3	Typen von Aktivierungsfunktionen	288
19.4	Das Perzeptron.	288
19.5	Deltaregel.	290
19.6	Das mehrschichtige Perzeptron	291
19.7	Klassifikation	293
19.8	Deep Learning	294
19.9	Backpropagation	294
19.10	Varianten der Gradientenverfahren	298
19.11	Ablauf des Backpropagation-Lernalgorithmus	298
19.12	Regression	300
19.13	Under- und Overfittung	301
19.14	Funktionsapproximation	302
19.15	Optimierung von KNN mit naturanalogen Optimierungsmethoden	304
19.16	Netztopologien.	307
19.17	Lernen mit künstlichen neuronalen Netzen	307
	Literatur.	308
20	Selbstorganisierende Karten	309
20.1	Kohonen-Karte.	309
20.2	Einführendes Beispiel (SOM-Algorithmus)	310
20.3	SOM-Lernalgorithmus.	311
20.4	Klassifizierung von Daten mit SOM	314
20.5	Lösung des TSP mit der SOM-Methode	315
	Literatur.	316

21	Hopfield-Netze	317
21.1	Definition eines Hopfield-Netzes	317
21.2	Aktualisierung in Hopfield-Netzen	318
21.3	Energiefunktion	319
21.4	Speichern von Mustern	320
21.5	Wiederherstellung eines Musters	322
21.6	Hopfield-Algorithmus mit dem Simulated-Annealing-Verfahren	323
21.7	Anwendungen auf Optimierungsprobleme	324
	Literatur	328
22	Reinforcement Learning	331
22.1	Einführung	331
22.2	Menace	332
22.3	Formale Beschreibung von RL	333
22.4	Markovsche Entscheidungsprozesse	335
22.5	Return	336
22.6	Policy	337
22.7	State-Value-Funktion	338
	22.7.1 Definition der State-Value-Funktion	338
	22.7.2 Beispiel der State-Value-Funktion	338
22.8	Action-Value-Funktion	339
22.9	Optimale Policies und optimale Value-Funktionen	340
22.10	Q-Learning	340
22.11	DQN-Algorithmus	343
22.12	REINFORCE-Algorithmus	346
	22.12.1 Policy Gradient Theorem	346
	22.12.2 REINFORCE Algorithmus	347
	22.12.3 Beispiel (Vier Gewinnt)	348
	22.12.4 REINFORCE mit Baseline	349
	Literatur	350
23	Optimierungsmethoden in Deep Learning	351
23.1	Problem der verschwindenden und explodierenden Gradienten	351
23.2	Gradient Clipping	352
23.3	Momentum-Optimierung	352
23.4	Exponentiell gleitender Durchschnitt	354
23.5	RMSProp-Optimierung	354
23.6	Adam-Optimierung	355
	Literatur	356
24	Neuronale Optimierung mit dem Pointer-Netzwerk	357
24.1	Rekurrente neuronale Netze	357
24.2	Einführende Beispiele von RNN	358
24.3	Mathematische Beschreibung des RNN	359

24.4	Backpropagation Through Time	360
24.5	LSTM-Zellen	360
24.6	Seq2seq	362
24.7	Softmax-Funktion	362
24.8	Attention-Mechanismus	363
24.9	Stochastische Policy für das TSP	365
24.10	Pointer-Netzwerk	366
24.11	Trainierbare Zielfunktion für das TSP	367
24.12	Baseline	369
24.13	REINFORCE-Algorithmus für das TSP	369
24.14	Varianten der REINFORCE-Methode für Optimierungsprobleme	370
	Literatur	371
25	Neuronale Optimierung mit Transformer	373
25.1	Transformer	373
25.1.1	Encoder	373
25.1.2	Decoder	379
25.2	Greedy Rollout Baseline	382
25.3	REINFORCE mit Greedy Rollout Baseline	383
25.4	Testphase	384
	Literatur	384
26	Optimierung mit graphischen neuronalen Netzen	385
26.1	Graphische neuronale Netze	385
26.2	Grapheneinbettung	387
26.3	S2V-Einbettung	387
26.4	TSP als Markovscher Entscheidungsprozess	388
26.5	Parametrisierung der Action-Value-Funktion	389
26.6	S2V-DQN-Algorithmus für TSP	390
26.7	Testphase	391
26.8	Weitere Optimierungsverfahren mit GNN	392
	Literatur	392
27	Programmbibliotheken für Machine Learning	393
27.1	Programmbibliothek TensorFlow	393
27.2	Weitere Programmbibliotheken	394
	Literatur	397
	Stichwortverzeichnis	403

Abkürzungsverzeichnis

ABC	Artificial Bee Colony Algorithm
ACS	Ant Colony System
Adam	Adaptive Moment Estimation
aiNet	Künstlicher Immunnetzwerk-Algorithmus
AIS	Künstliches Immunsystem
AS	Ameisenalgorithmus
ASrank	Ranked-Based Ant System
AS-SCSP	Ant System-Shortest Common Supersequence Problem
BA	Bienenalgorithmus
BAT	Fledermausalgorithmus
BFD	Best Fit decreasing
BinBA	Binärer Fledermausalgorithmus
BN	Batch-Normalisierung
BPTT	Backpropagation through Time
C&P	Cutting and Packing Problem
CLONALG	Klonaler Selektions-Algorithmus
CVRP	Capacitated Vehicle Routing Problem
DPSO	Diskreter Partikelschwarmalgorithmus
EAS	Elitist Ant System
ERX	Edge Recombination Crossover
ES	Evolutionsstrategien
FFD	First Fit decreasing
GA	Genetischer Algorithmus
GAP	Generalised Assignment Problem
GNN	Graphische Neuronale Netze
HopSA	Hopfield-Simulated-Annealing-Algorithmus
KNN	Künstliche neuronale Netze
LSTM	Long Short-Term Memory
MDVVRP	Multi-Depot Vehicle Routing Problem
MMAS	MAX-MIN Ant System
MOBA	Multikriterieller Fledermausalgorithmus

MOPSO	Multikriterielle Partikelschwarmoptimierung
NCO	Neuronale kombinatorische Optimierung
NFD	Next Fit decreasing
NSA	Negativer-Selektions-Algorithmus
NSGAI	Nondominated Sorting Genetic Algorithm II
OX	Order Crossover
PMX	Partially Matched Crossover
PSA	Positiver-Selektions-Algorithmus
PSO	Partikelschwarmalgorithmus
RL	Reinforcement Learning
RMSProp	Root Mean Square Propagation
RNN	Rekurrente Neuronale Netze
S2V	Structure2Vec
SA	Simulated Annealing
SCSP	Shortest Common Supersequence Problem
SGD	stochastisches Gradientenverfahren
SI	Sinflut-Algorithmus
SUS	Stochastic Universal Sampling
TA	Threshold-Accepting-Algorithmus
TSP	Traveling Salesman Problem
VPR	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows

Teil I

Optimierungsprobleme

Kapitel 1

Einführung



Es gibt eine unübersehbare Vielzahl verschiedener Optimierungsprobleme in der realen Welt, wie zum Beispiel Optimierungsaufgaben aus den Bereichen Logistik, Technik, Finanzwirtschaft, Medizin, Telekommunikation oder Verkehrsplanung. Im ersten Kapitel werden verschiedene Optimierungsprobleme aus diesen Bereichen aufgelistet. Weiterhin werden Grundbegriffe der Optimierung eingeführt.

1.1 Optimierungsprobleme in der realen Welt

Im Folgenden werden stellvertretend einige Optimierungsprobleme aus unterschiedlichen Bereichen aufgeführt:

Produktionsplanung

- Minimierung des Verschnitts beim Zuschneiden von Rohren, Holzplatten oder Stoffen und beim Stanzen von Blechen
- Minimierung des Laderaums beim Beladen von LKWs, Containern oder Schiffen
- Optimierung der Lagerhaltung einer Firma
- Kostengünstige Mischung aus einem Vorrat (z. B. Futtermischungen)

Tourenplanung

- Minimierung der Wegstrecke zwischen zwei Orten
- Bestimmung der kürzesten Rundreise
- Optimierung der Routen für Müllabfuhr, Wartungsfahrten, Paketlieferung oder Entleerung von Briefkästen
- Optimierung der Laufwege von Bohrautomaten
- Optimierung der Auslastung von Fahrzeugen

Städte- und Verkehrsplanung

- Optimale Steuerung des Verkehrsflusses
- Optimierung bei der Auslegung von Strom- und Wasserleitungen

- Optimierung bei der Standortbestimmung wie zum Beispiel von Feuerwehrationen und Windrädern
- Optimale Platzierung von Parkplätzen

Telekommunikation

- Optimale Ausrichtung von Antennen
- Optimierung der Auslegung von Telefonleitungen
- Suche nach der besten Standortverteilung für Mobilfunkmasten
- Optimale Steuerung von Datenströmen in Kommunikationsnetzwerken während des Betriebs

Ablaufplanung

- Optimierung von Arbeitsabläufen
- Optimale Ausnutzung von Maschinen
- Maximierung der Kapazitätsauslastung
- Minimierung der Terminabweichung
- Optimierung bei der Erstellung von Stunden-, Bahn-, Bus- und Flugplänen

Medizin

- Maximierung der Wirkung eines Medikaments bei gleichzeitiger Minimierung der Nebenwirkungen
- Optimierung bei der medizinischen Bildverarbeitung in der Radiologie
- Erstellung eines optimalen Diätplans

Technik

- Optimierung der Oberfläche eines PKWs zur Reduzierung des Luftwiderstandsbeiwertes
- Optimale Kalibrierung eines Motors zur Minimierung des Kraftstoffverbrauchs und des Schadstoffausstoßes bei möglichst hoher Motorleistung
- Optimierung von PKW-Reifen
- Minimierung der Laufwege von Schweißrobotern beim Punktschweißen von Karosserieteilen
- Optimierung von Flugzeugflügeln
- Optimierung von Brillengläsern

Finanzwirtschaft

- Portfolio-Optimierung
- Risikominimierung von Finanzprodukten
- Optimierung von Geschäftsabläufen
- Optimierung von Investitionen
- Preisoptimierung
- Kostenminimierung

1.2 Grundbegriffe der Optimierung

Bei einer Optimierung geht es darum, zu einer Bewertungsfunktion $f : S \rightarrow \mathbb{R}$ unter allen Elementen aus einer Menge S ein Element mit der besten Bewertung zu ermitteln. Die zu optimierende Funktion f wird *Zielfunktion* und im Zusammenhang mit evolutionären Algorithmen (Kap. 12) auch *Fitnessfunktion* genannt. Der Definitionsbereich S von f wird mit *Suchraum* bezeichnet, die Elemente aus dem Suchraum nennt man *Lösungen*.

1.2.1 Globales und lokales Optimum

Definitionen

- Eine Zielfunktion $f : S \rightarrow \mathbb{R}$ besitzt in $x^* \in S$ ein *globales Maximum* bzw. *globales Minimum*, wenn für alle $x \in S$ gilt: $f(x) \leq f(x^*)$ bzw. $f(x^*) \leq f(x)$.
- Eine Zielfunktion $f : S \rightarrow \mathbb{R}$ besitzt in $x^* \in S$ ein *lokales Maximum* bzw. *lokales Minimum*, wenn es eine (kleine) Umgebung U von x^* gibt, sodass für alle $x \in U \cap S$ gilt: $f(x) \leq f(x^*)$ bzw. $f(x^*) \leq f(x)$ (s. Abb. 1.1).

1.2.2 Umwandlung eines Maximierungsproblems in ein Minimierungsproblem und umgekehrt

Durch Multiplikation einer Zielfunktion $f(x)$ mit -1 kann ein Maximierungsproblem (Minimierungsproblem) in ein Minimierungsproblem (Maximierungsproblem) umgewandelt werden. Die Extremstellen beider Funktionen sind identisch (s. Abb. 1.2).

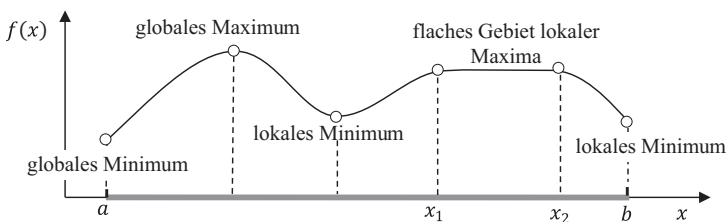
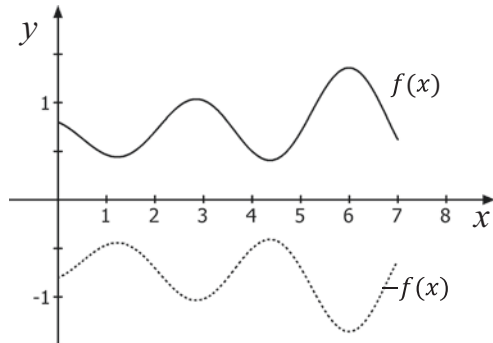


Abb. 1.1 Globale und lokale Extremwerte der Zielfunktion $f : [a, b] \rightarrow \mathbb{R}$ mit den Randextremwerten in a und b . In dem Bereich zwischen x_1 und x_2 ist jeder Punkt eine Maximalstelle.

Abb. 1.2 Graphisch bewirkt die Multiplikation der Zielfunktion $f(x)$ mit -1 eine Spiegelung an der x -Achse. Die Lage der Extremstellen bleibt unberührt.



1.2.3 Notationen

Für die Bestimmung eines Maximums bzw. Minimums der Funktion $f(x)$ ist folgende Schreibweise üblich:

$$f(x) \rightarrow \max \quad \text{bzw.} \quad f(x) \rightarrow \min.$$

Besitzt die Funktion $f: S \rightarrow \mathbb{R}$ in x^* ein globales Maximum bzw. Minimum, so schreibt man

$$f(x^*) = \max_{x \in S} f(x) \quad \text{bzw.} \quad f(x^*) = \min_{x \in S} f(x)$$

und

$$x^* = \operatorname{argmax}_{x \in S} f(x) \quad \text{bzw.} \quad x^* = \operatorname{argmin}_{x \in S} f(x).$$

1.2.4 Supremum und Infimum

Supremum Ist die Funktion $f: S \rightarrow \mathbb{R}$ nach oben beschränkt, so existiert die kleinste obere Schranke M mit $f(x) \leq M$ für alle $x \in S$. M heißt *Supremum* von $f(x)$ und man schreibt

$$M = \sup_{x \in S} f(x).$$

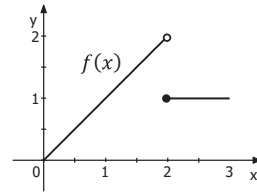
Infimum Ist die Funktion $f: S \rightarrow \mathbb{R}$ nach unten beschränkt, so existiert die größte untere Schranke N . N heißt *Infimum* von $f(x)$ und man schreibt

$$N = \inf_{x \in S} f(x).$$

Beispiel:

Gegeben sei die Funktion

$$f: [0,3] \rightarrow \mathbb{R}, f(x) = \begin{cases} x & \text{wenn } x \in [0,2) \\ 1 & \text{wenn } x \in [2,3] \end{cases}$$



Die Funktion $f(x)$ besitzt im Intervall $[0,3]$ in 0 ein globales Minimum und kein Maximum. Es gilt

$$f(0) = \min_{x \in [0,3]} f(x) = \inf_{x \in [0,3]} f(x) = 0$$

und

$$\sup_{x \in [0,3]} f(x) = 2.$$

Die Funktion $f(x)$ nimmt im Intervall $[0,3]$ ihr Supremum nicht an.

1.2.5 *Kontinuierliche, diskrete und kombinatorische Optimierung*

Ein Optimierungsproblem mit der Zielfunktion $f: S \rightarrow \mathbb{R}$ heißt

- **kontinuierlich**, wenn der Suchraum S überabzählbar unendlich ist (z. B. $S = [a,b] \subset \mathbb{R}$).
- **diskret**, wenn S endlich oder abzählbar unendlich ist (z. B. $S = \text{Menge } \mathbb{Z} \text{ der ganzen Zahlen}$).
- **kombinatorisch**, wenn S endlich ist (z. B. $S = \text{Menge der Zahlen } 1, \dots, 10$).

Kapitel 2

Kontinuierliche Optimierungsprobleme



In diesem Kapitel werden Beispiele von kontinuierlichen Optimierungsproblemen vorgestellt. Weiterhin wird der Begriff Regression eingeführt. Bei der Regression wird eine Funktion vom bestimmten Typ an eine Datenwolke von Punkten angepasst, wobei man von linearer, quadratischer und exponentieller Regression spricht, wenn die anzupassende Funktion linear, quadratisch bzw. exponentiell ist. Weiterhin werden Extremwerteigenschaften von konkaven und konvexen Funktionen beschrieben.

2.1 Graph einer Funktion

Im Folgenden sei $f : S \rightarrow \mathbb{R}$ eine zu optimierende kontinuierliche Funktion, wobei S eine Teilmenge von \mathbb{R}^n ist.

- Für eine Funktion $f : S \rightarrow \mathbb{R}$ mit $S \subset \mathbb{R}$ heißt die Menge $\{(x, f(x)) \in \mathbb{R}^2 : x \in S\}$ *Graph* von f und stellt eine ebene Kurve dar.
- Für $S \subset \mathbb{R}^2$ beschreibt der Graph $\{(x, y, f(x)) \in \mathbb{R}^3 : (x, y) \in S\}$ ein Gebirge, in dem die Talsohlen die lokalen Minima und die Berggipfel die lokalen Maxima darstellen. Der höchste Berg ist dann das globale Maximum und das tiefste Tal das globale Minimum.

Beispiele

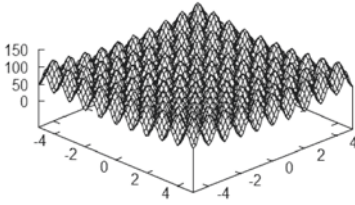
Im Folgenden sind die Funktionsgebirge von vier (Benchmark)-Funktionen dargestellt, die oft für Performance-Tests von Optimierungsalgorithmen verwendet werden (s. Abb. 2.1).

Rastrigin-Funktion Diese Funktion ist definiert durch

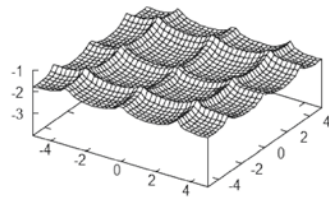
$$f(x, y) = 20 + (x^2 - 10 \cos(2\pi x)) + (y^2 - 10 \cos(2\pi y)).$$

Sie stellt ein schweres Optimierungsproblem dar, da die Wertelandschaft sehr viele lokale Minima und ein globales Minimum im Punkt (0,0) besitzt.

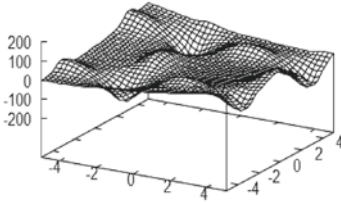
Rastrigin-Funktion



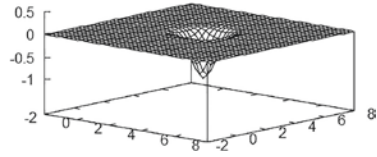
Cross-Tray-Funktion



Egg Crate Funktion



Easom's Funktion

**Abb. 2.1** Die Graphen der vier Benchmark-Funktionen

Cross-Tray-Funktion Diese Funktion ist definiert durch

$$f(x, y) = -0.0001 \left| \sin(x) \sin(y) \exp \left(\left| 100 - \frac{1}{\pi} \sqrt{x^2 + y^2} \right| \right) \right| + 1)^{0.1}.$$

Die vier globalen Minima dieser Funktion sind gegeben durch $(x_0, y_0) = (\pm 1.349407, \mp 1.349407)$ mit $f(x_0, y_0) = -2.0626122$.

Egg-Crate-Funktion Diese Funktion ist definiert durch

$$f(x, y) = x^2 + y^2 + 25(\sin^2 x + \sin^2 y).$$

Das globale Minimum liegt in $(0, 0)$, wobei $f(0, 0) = 0$.

Easom's Funktion Diese Funktion ist definiert durch

$$f(x, y) = -\cos(x) \cos(y) \exp(-(x - \pi)^2 - (y - \pi)^2).$$

Sie besitzt in (π, π) ein globales Minimum mit $f(\pi, \pi) = -1$.

2.2 Optimierungen mit Nebenbedingungen

In vielen Anwendungen sucht man Extremstellen einer Funktion $f: S \rightarrow \mathbb{R}$, $S \subset \mathbb{R}^n$, wobei die Lösungen x zusätzlich die Nebenbedingung $x \in M$ erfüllen sollen. Die Menge $M \cap S$ heißt dann *zulässiger Bereich* und $x \in M \cap S$ *zulässige*

Lösung. Die Nebenbedingungen können in Form von Ungleichungen $g(x) \geq 0$ oder Gleichungen der Form $h(x) = 0$ gegeben sein.

Optimierungsproblem mit Nebenbedingungen

Ein Optimierungsproblem ist gegeben durch

Zielfunktion: $f(x) \rightarrow \min(\max)$

Nebenbedingungen: $g_i(x) \geq 0, \quad i = 1, \dots, m$

$h_k(x) = 0, \quad k = 1, \dots, n$

$x \in S$

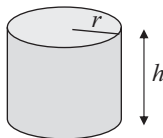
Ungleichungen der Form $g(x) \leq 0$ können durch Vorzeichenwechsel in eine Ungleichung ≥ 0 übergeführt werden.

2.3 Anwendungsbeispiele

• Optimierungsproblem aus der Wirtschaft

Zu bestimmen sind die Maße einer zylindrischen Dose mit einem Volumeninhalt von V , sodass die Dose mit minimalem Material hergestellt werden kann.

Die zu minimierende Oberfläche O ist gegeben durch die Summe des Flächeninhalts der Mantelfläche $2\pi rh$ und des doppelten Flächeninhalts der Deckelfläche πr^2 . Das Volumen der zylindrischen Dose ergibt sich aus $V = \pi r^2 h$.



Das Optimierungsproblem lautet demzufolge:

Zielfunktion: $f(h, r) = 2\pi r^2 + 2\pi rh \rightarrow \min$
 $\pi r^2 h = V$

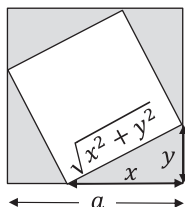
Nebenbedingungen: $r \geq 0$
 $h \geq 0$

Die Lösung des Optimierungsproblems kann mit den in Kap. 6 beschriebenen analytischen Methoden leicht berechnet werden. Die optimale Lösung lautet (vgl. Abschn. 6.6.7)

$$r_0 = \sqrt[3]{\frac{V}{2\pi}}, \quad h_0 = 2\sqrt[3]{\frac{V}{2\pi}}.$$

• Optimierungsproblem aus der Geometrie

In einem Quadrat mit einer gegebenen Seitenlänge a soll ein Quadrat so eingeschrieben werden, dass dessen Flächeninhalt minimal ist.



Einbeschriebenes Quadrat in einem Quadrat

Der Flächeninhalt des eingeschriebenen Quadrats ist gleich $x^2 + y^2$. Man erhält damit folgendes Optimierungsproblem:

Zielfunktion: $f(x, y) = x^2 + y^2 \rightarrow \min$
 $x + y = a$

Nebenbedingungen: $a - x \geq 0$
 $a - y \geq 0$
 $x, y \geq 0$

Mittels analytischer Methoden (vgl. Kap. 6) erhält man als optimale Lösung:

$$x = y = \frac{a}{2}.$$

• Regression

Einführendes Beispiel Eine Feder wird durch Anhängen eines Gewichts gedehnt. Nach dem Hookschen Gesetz besteht der lineare Zusammenhang $F = D \cdot x$ zwischen der Auslenkung x und der Gewichtskraft F , wobei D die Federkonstante ist. Das Hooksche Gesetz ist zum Beispiel auch auf die Ausdehnung eines Gummiseils beim Bungee-Springen anwendbar, wobei die Ausdehnung von der Gummihärte D und vom Gewicht des Bungee-Springers abhängt. Für die Bestimmung der Federkonstante D einer Feder seien die fünf Messdaten wie in Abb. 2.2 ermittelt worden. Es ist das Ziel, die Gerade $F = D \cdot x$ bestmöglich an die Messdaten anzupassen. Diese Gerade nennt man Ausgleichsgerade. Hierzu wird die Summe der Quadrate der Fehler $F_i - Dx_i$ (senkrechte Abweichung zur Geraden) minimiert:

$$f(D) = \sum_{i=1}^5 (F_i - Dx_i)^2 \rightarrow \min.$$

Das Quadrieren ist erforderlich, da bei der einfachen Summierung die negativen und positiven Fehler sich gegenseitig aufheben können. Die Verwendung von Beträgen ist nicht sinnvoll, da das Rechnen mit Beträgen aufwendig ist.

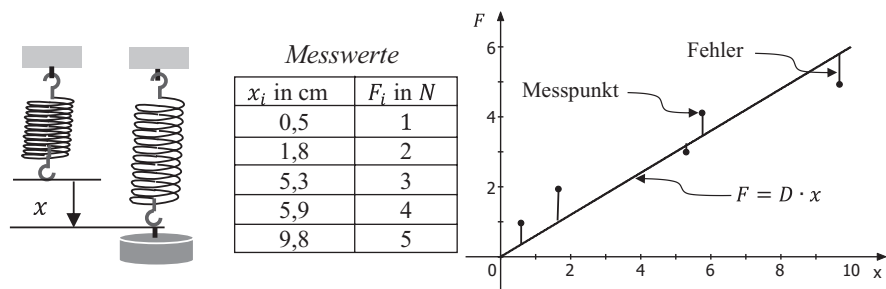


Abb. 2.2 Bestimmung einer Federkonstante D durch eine Ausgleichsgerade $F = D \cdot x$, die optimal an die Messdaten anzupassen ist

Arten von Regressionen

Bei der Regression wird eine Funktion $g(x)$ vom bestimmten Typ optimal an eine Datenwolke von Punkten $(x_i, y_i), i = 1, \dots, n$, angepasst (s. Abb. 2.3). Je nach Funktionstyp $g(x)$ unterscheidet man folgende Arten von Regressionen:

Lineare Regression ($g(x) = ax + b$):

$$f(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2 \rightarrow \min$$

Quadratische Regression ($g(x) = ax^2 + bx + c$):

$$f(a, b, c) = \sum_{i=1}^n (y_i - (ax_i^2 + bx_i + c))^2 \rightarrow \min$$

Exponentielle Regression ($g(x) = a \cdot e^{kx}$):

$$f(a, k) = \sum_{i=1}^n (y_i - ae^{kx})^2 \rightarrow \min$$

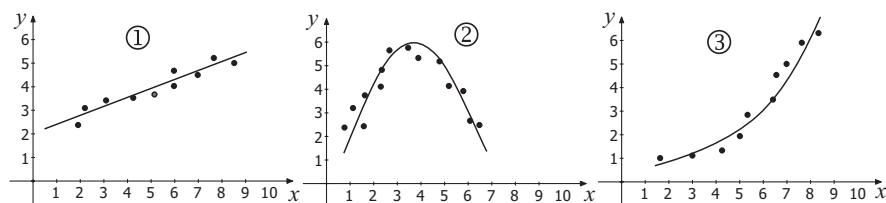


Abb. 2.3 ① Lineare Regression: Anpassung durch eine Gerade ② Quadratische Regression: Anpassung durch eine Parabel ③ Exponentielle Regression: Anpassung durch eine Exponentialfunktion