

OPTIMIZAREA MATEMATICĂ

În căutarea celei mai bune opțiuni

LUCAS LACASA

Traducere de Elena-Anca Coman

CUPRINS

Capitolul 0

Richard Ernest Bellman	9
RAND Corporation	11
Programare dinamică	13

Capitolul 1

Ce este optimizarea matematică?	17
--	----

Capitolul 2

Taxonomia problemelor de optimizare	23
Optimizare convexă vs. neconvexă	23
Programare liniară	26
<i>Exemplul 1</i>	27
<i>Exemplul 2</i>	29
Programare cu numere întregi	30
Programare neliniară	31
Optimizare combinatorie	32
<i>Exemplul 3</i>	34
Calcul variațional	36
Control optim	37
<i>Problema brahistocronei</i>	38
Optimizare stocastică	42
<i>Exemplul 4</i>	43

Capitolul 3

Metode de rezolvare analitice

Ω continuu, funcție de cost continuă: în căutarea minimului unei funcții	47
<i>Exemplul 5 ($n = 1$)</i>	48
<i>Exemplul 6 ($n = 2$)</i>	50
Funcții obiectiv continue cu restricții: metoda multiplicatorilor Lagrange	51
<i>Exemplul 7 ($m = 2$), ($s = 1$)</i>	53
Calculul variațiilor: ecuația Euler-Lagrange	54
<i>Cea mai scurtă cale între două puncte</i>	55
	56

Capitolul 4

Algoritmi de calcul: metode exacte, metode iterative și metode euristice

Algoritm simplex	59
<i>Ce este un politop?</i>	61
<i>Exemplul 8</i>	64
Programare dinamică: cu grijă, forța brută devine eficientă	66
<i>Exemplul 9. Seria Fibonacci</i>	69
<i>Exemplul 10. Cea mai scurtă cale între două noduri ale unui graf</i>	71
Optimizare locală: hill climbing și metoda gradientului	75
<i>Problema comis-voiajorului</i>	78
Simulated annealing (Călirea stimulată)	80
Simulated annealing (Călirea stimulată) și arta întăririi oțelului unei săbii	83
Algoritmi genetici	84
<i>Problema ghiozdanului</i>	86
	90

Capitolul 5

Aplicații

Căutarea tiparelor în date: învățare statistică	93
	94

Rețele neuronale și învățare profundă	101
<i>Succese recente ale tehnologiei bazate pe învățare profundă</i>	120
Fizica este o mare problemă de optimizare	123
Fizica este o mare problemă de optimizare (din nou): termodinamica	126
Concluzii	131
Anexă	133
Bibliografie recomandată	141

Capitolul 1

Ce este optimizarea matematică?

*Căutați, comparați și, dacă găsiți
ceva mai bun, cumpărați-l.*
(Manuel Luque, detergenți Colon)

Optimizarea înseamnă a găsi cea mai bună opțiune, dintr-un ansamblu de posibilități, conform unui anumit criteriu. Pentru a explica despre ce este vorba și pentru a arăta modul în care sunt formulate matematic problemele de optimizare, să folosim un exemplu.

Măine începe vacanța și plecăm într-o călătorie cu mașina: plecăm din Madrid pentru a ajunge în Pirineii catalani. Cum mergem? Ce traseu alegem? Aceasta este o problemă de optimizare și, în mod normal, calea optimă va depinde de criteriul pe care dorim să-l optimizăm. Poate că vrem să ajungem cât mai curând posibil, deoarece nu ne place să conducem: criteriul nostru de optimizare este timpul (în special, vrem să *minimizăm* timpul). Sau poate ne interesează mediul înconjurător (și buzunarul nostru) și dorim să alegem ruta care consumă mai puțină benzină. În acest caz, traseul optim nu este neapărat cel mai rapid! (consumul de benzină nu crește liniar cu viteza). Dacă, pe de altă parte, problema este pur economică, probabil că preferăm să economisim cât mai mult pe taxele de drum, fără să ne pese de timpul necesar pentru a ajunge la destinație. În cele din urmă, un alt criteriu ar putea fi alegerea celui mai frumos traseu, acela care ne oferă cele mai frumoase priveliști: în acest din urmă caz, dorim să *maximizăm* calitatea estetică. De observat, cititorule, că, în prezent,

aplicațiile cunoscute de toată lumea, precum Google Maps, cu greu ne oferă cel mai rapid traseu, dar există și alte platforme care optimizează alte tipuri de criterii. De fapt, concepte precum maximizarea calității estetice – care pot suna vag și abstract – sunt cercetate în prezent de oameni de știință din departamentele de inovare ale centrelor tehnologice precum Google sau Nokia Bell Labs, unde încearcă să dezvolte softuri care să răspundă la întrebări precum: în orașul meu, ce rută ar trebui să aleg ca să merg de la A la B pentru a (i) fi expus la cel mai redus zgomot de mediu; (ii) pentru a fi expus la mai puțină poluare și pentru a respira mai mult aer curat; (iii) pentru a mă bucura de o arhitectură de calitate superioară etc.?

Prin urmare, matematic am putea spune că o problemă de optimizare poate fi formulată astfel: «Din mulțimea de posibilități Ω , găsește-o pe aceea care MINIMIZEAZĂ criteriul C », sau

$$\min_{x \in \Omega} C(x).$$

Cititorul trebuie să observe că, în cazul în care ceea ce ne dorim este a *maximiza* în loc de a *minimiza* funcția noastră obiectiv C , trebuie doar să punem un „minus” formulei anterioare, deoarece a *maximiza* C este echivalent din punct de vedere matematic cu a *minimiza* $-C$ (ca și cum am spune că cel mai bogat este cel mai puțin sărac). În această carte vom folosi alternativ minimizarea sau maximizarea ca formă canonică de optimizare, în funcție de context. În exemplul concret pe care îl analizăm, Ω este domeniul tuturor rutelor posibile care leagă Madridul de Pirineii catalani, iar C este criteriul ales. Dacă, de exemplu, definim C ca „timp estimat al traseului”, atunci dat fiind un element x al domeniului Ω (o rută care unește Madridul de Pirineii), $C(x)$ indică un număr real pozitiv (timpul estimat pentru parcurgerea acelei rute, să zicem, în ore).

Soluția optimă, pe care o putem denumi x_{opt} , este aceea pentru care $C(x_{opt}) < C(x)$, pentru toate acele elemente $x \in \Omega$ diferite de x_{opt} . Cum găsim x_{opt} ? Astfel, direct, o strategie de forță brută ar putea fi: ei

bine, dacă există puține rute posibile (dacă mulțimea Ω are un *cardinal* $|\Omega|$ jos), atunci putem verifica, una câte una, toate rutele posibile, putem calcula timpul pentru fiecare și o putem alege pe cea mai rapidă. Problema forței brute este că, atunci când numărul traseelor pe care trebuie să le verificăm este prea mare (matematic, acest lucru se scrie: $|\Omega| \gg 1$), probabil ne va lua prea mult timp pentru a o găsi pe cea optimă. Avem nevoie de un răspuns până mâine!

Și, bineînțeles, de la Madrid la Pirinei, privind harta rutieră, există nenumărate posibilități! Dar, atenție, dintr-o lovitură am putea elimina multe dintre ele; de exemplu, cu siguranță nu ne interesează rutele care folosesc drumuri locale sau județene, ci doar șoselele și autostrăzile. Prin această cerință, ceea ce facem este să reducem mulțimea noastră inițială Ω la alta, într-adevăr, mai mică. În mod formal, dacă denumim h numărul total de drumuri locale și județene pe care le conține o posibilă rută, putem scrie:

$$\min_{x \in \Omega} C(x)$$

dacă $h = 0$.

Tocmai am scris o problemă clasică de optimizare *cu restricții*. În exemplul pe care l-am discutat, spațiul Ω este *discret*, adică este o mulțime finită de elemente (rutele). Restricția pe care am impus-o este că acele rute care implică drumuri locale și județene nu vor fi *soluții fezabile*. Prin urmare, problema se reduce la găsirea rutei care necesită mai puțin timp doar dintre cele *fezabile*.

Există multe alte probleme de optimizare unde Ω nu este un domeniu discret, ci este continuu. Poate fi, de asemenea, un domeniu infinit sau finit, poate avea una sau mai multe dimensiuni, poate fi măsurabil sau nu etc. Criteriul C este denumit în mod obișnuit „funcție de cost” sau „funcție obiectiv”, iar aceasta, la rândul ei, poate avea proprietăți foarte diferite, în funcție de caz. În cele din urmă, posibilele restricții la care supunem problema de optimizare pot fi de multe tipuri (egalități, inegalități etc.). Optimizarea matematică este

o ramură a matematicii aplicate care se ocupă de formularea în modul cel mai general și mai abstract posibil a problemei pe care tocmai am descris-o, precum și de proiectarea metodelor care să ne ajute să găsim soluția optimă. După cum vom vedea în capitolele următoare, există o întreagă taxonomie a tipurilor de probleme de optimizare, suficient de mare încât să fie necesar să se elaboreze diferite teorii și instrumente de calcul pentru a aborda fiecare dintre tipuri. În special, rețineți că pentru a defini o problemă de optimizare trebuie să descriem:

- Spațiu de soluții Ω : acest spațiu poate fi discret sau continuu. Elementele $x \in \Omega$ pot fi scalare (numere) binare, întregi sau reale, vectori de dimensiune finită sau dimensiune infinită, sau chiar funcții (calculul variațiilor) sau ecuații (ca în cazul controlului optim).
- Funcție de cost $C(x)$: poate fi o funcție continuă sau discretă, diferențiabilă sau nu, de una sau mai multe dimensiuni. În aplicații precum inteligența artificială, există sisteme în care funcția de cost poate avea o dimensiune foarte mare (milioane).
- Restricții suplimentare: pot fi egalități sau inegalități sau condiții care, în general, reduc spațiul pentru soluții fezabile.

Fiecare tip de problemă de optimizare, așa cum vom vedea, este asociat cu un tip de caracteristică pentru spațiul Ω , funcția de cost $C(x)$ și restricțiile suplimentare. În funcție de caz, ne vom întâlni cu domenii precum *programarea liniară*, *optimizarea combinatorie*, *calculul variațiilor*, *controlul optim*... Fiecare dintre aceste categorii este rodul muncii combinate a unui număr mare de oameni de știință din ultimele secole. În unele cazuri există metode matematice care oferă soluția exactă (metode analitice) sau există algoritmi care calculează optimul într-un număr de pași. Pentru multe alte cazuri nu există metode exacte care să găsească optimul și va trebui să recurgem la algoritmi euristici care propun strategii de căutare inteligente.

În următoarele capitole vom învăța să clasificăm întregul ansamblu de probleme de optimizare existente, vom înțelege cum putem rezolva aceste probleme cu diferite tehnici matematice și de simulare computerizată și vom explora utilizarea acestor instrumente în sarcini foarte diverse, precum simularea de sisteme fizice, inteligența artificială și învățarea automată sau planificarea.

Pregătiți-vă, începem!

Capitolul 2

Taxonomia problemelor de optimizare

Optimizare convexă vs. neconvexă

Înainte de a începe, este necesar să definim două concepte care ne vor fi foarte utile la clasificarea tipurilor de probleme de optimizare: domeniile și funcțiile convexe.

Domeniu convex. Îi vom cere cititorului să decupeze cu foarfeca o bucată de carton cu orice formă și să arunce restul. Pe cartonul rămas,

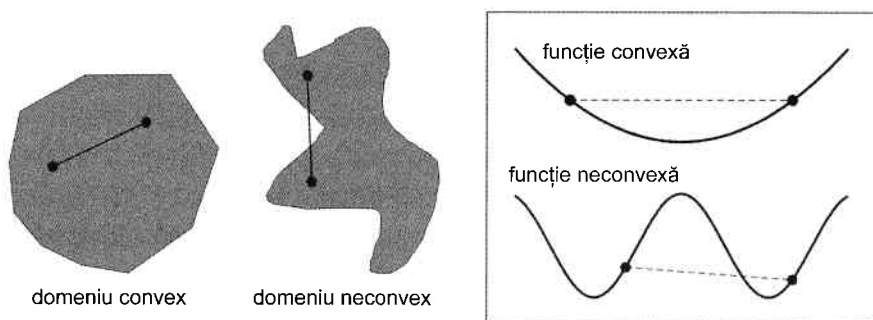


Figura 2. Convexitatea domeniilor și funcțiilor.

desenați două puncte și încercați să trasați o linie care să unească cele două puncte. Este posibil să trasați linia fără ca aceasta să iasă din carton? Dacă este posibil, desenați din nou două puncte diferite. De data aceasta, ați putut să le uniți cu o linie dreaptă? Mai general, luând oricare două puncte pe acel carton, credeți că linia care le unește va fi întotdeauna trasată în interiorul cartonului? Dacă da, cartonul pe care l-ați construit se numește „domeniu convex”.

Într-un mod mai riguros, un domeniu de elemente Ω este convex dacă, pentru orice pereche de elemente (x, x') din domeniu, toate punctele segmentului care unește x cu x' sunt, la rândul lor, elemente ale domeniului Ω . Dimpotrivă, în cazul în care există cel puțin două elemente ale unui domeniu a căror linie de unire conține un punct extern domeniului (adică dacă la un moment dat iese în afara conturului), atunci domeniul este neconvex.

Funcție convexă. Există o proprietate de convexitate asociată funcțiilor. În limbaj colocvial, vom spune că o funcție este convexă dacă, pentru oricare două puncte (x, x') , curba care reprezintă acea funcție evaluată între x și x' rămâne mereu *sub* segmentul care unește valoarea funcției în puncte. Matematic vorbind, acest lucru este exprimat după cum urmează: să definim un parametru auxiliar $t \in [0,1]$. Expresia $(1-t)x + tx'$ se spune că *interpolează liniar* x și x' , întrucât pentru $t = 0$ acea expresie se reduce la x , pentru $t = 1$ expresia ia valoarea x' , iar pentru orice valoare a lui t între 0 și 1, expresia reprezintă coordonata unui punct de pe segmentul $[x, x']$. Ei bine, o funcție $f(x)$ este definită drept convexă dacă pentru fiecare pereche de puncte x, x' și pentru $t \in [0,1]$ se îndeplinește

$$\underbrace{f((1-t)x + tx')}_{\text{străbate curba între } f(x) \text{ și } f(x')} \leq \underbrace{(1-t)f(x) + tf(x')}_{\text{segment care unește } f(x) \text{ cu } f(x')} .$$

Ca un comentariu suplimentar, să observe cititorul că, dacă curba unei funcții între două puncte este sistematic *mai mare* decât valoarea segmentului care le unește, funcția va fi *concavă* în loc de *convexă*. Dacă o funcție $f(x)$ este convexă, atunci $-f(x)$ este concavă și invers.

Echipați cu aceste noțiuni, putem începe. Prima întrebare care trebuie pusă, atunci când ne confruntăm cu o problemă de optimizare, este dacă această problemă are *una sau mai multe* soluții optime. De exemplu, să analizăm problema minimizării unei funcții. Este posibil ca această funcție să aibă minime *locale*, adică puncte care sunt minime la nivel local. În aceste cazuri, găsirea minimului global va fi mai dificilă decât în cazul în care funcția are un minim unic.

Matematic vorbind, dacă funcția obiectiv C de minimizat este *convexă* și, în plus, spațiul de căutare Ω e un *domeniu convex*, atunci problema de optimizare este convexă. Această mare clasă de probleme are particularitatea de a avea un minim unic, care este, prin urmare, minimul global. Aceasta înseamnă că, în cazul în care am

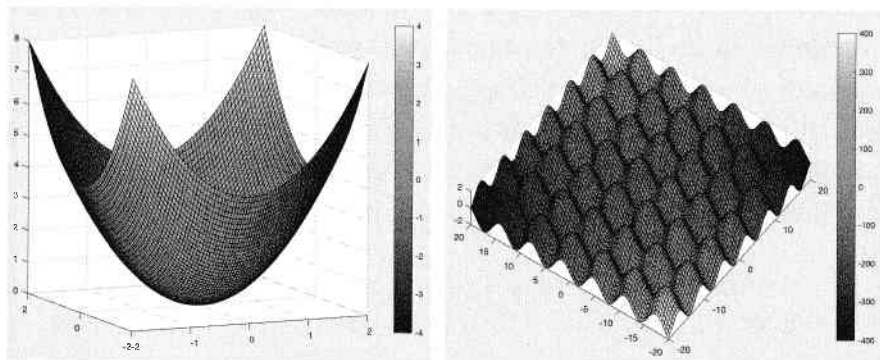


Figura 3. Stânga: exemplu de funcție obiectiv $C(x_1, x_2)$ convexă, cu un singur minim global. Dreapta: exemplu de funcție obiectiv neconvexă, cu o multitudine de minime, în care găsirea minimului global este mai dificilă.

putea găsi un punct în spațiu care să fie cel puțin un minim local al funcției obiectiv, adică într-un mediu restrâns al acelui punct funcția obiectiv să fie minimă, vom putea garanta că acel punct este soluția la problema noastră fără să fie nevoie să verificăm dacă există alte minime locale.

Celălalt grup mare de probleme, în care fie funcția de minimizat nu este convexă, fie spațiul de căutare nu este convex, se alătură problemelor de optimizare neconvexe. În această clasă, în general, funcția de cost poate avea multiple minime locale. Astfel, *a priori* nu se poate garanta care va fi minimul global (soluția problemei) decât dacă se pot găsi în mod sistematic toate minimele locale, pentru a compara apoi care dintre ele este cel mai mic.

În secțiunile următoare vom enumera și discuta cele mai importante clase de probleme de optimizare.

Programare liniară

În această primă parte vom formula ceea ce se cunoaște drept „probleme de programare liniară”, denumite și „probleme de optimizare liniară”. Cititorul trebuie să știe că termenul *programare* nu trebuie confundat cu un limbaj de programare: sensul în acest context este sinonim cu *optimizare* și, de fapt, *programare matematică* este un sinonim oarecum învechit al expresiei *optimizare matematică*. Într-o problemă de programare liniară, funcția de cost C este o *funcție liniară* (adică argumentele sale apar liniar în expresia funcției):

$$C(x_1, x_2, \dots, x_m) = c_1x_1 + c_2x_2 + \dots + c_mx_m,$$

unde $c_1, c_2, \dots, c_m$ sunt constante. În plus, restricțiile pe care le impunem sunt, la rândul lor, ecuații și inegalități *liniare*. În forma sa canonică, o problemă clasică de programare liniară cu m variabile și n restricții se scrie astfel:

a maximiza $C(x_1, x_2, \dots, x_m)$
supus la $A\bar{x} \leq b$
și $\bar{x} \geq \bar{0}$,

unde $\bar{x}$ este un vector de variabile de determinat $\bar{x} = (x_1, x_2, \dots, x_m)$, A este o matrice $n \times m$ de coeficienți constanți cunoscută și $\bar{b} = (b_1, b_2, \dots, b_n)$ este un vector de coeficienți cunoscut. Deoarece restricțiile sunt inegalități *liniare* (granițele sunt plane sau hiperplane), ele restricționează spațiul inițial al soluțiilor posibile la o formă geometrică numită „politop convex” (vom vorbi despre politopi în capitolul 4).

Această clasă de probleme de optimizare a fost inițial formulată și rezolvată de economistul sovietic Leonid Kantorovich și, în mod independent, de contemporanul său american T.C. Koopmans. Câțiva ani mai târziu, Kantorovich și Koopmans aveau să împartă Premiul Nobel pentru Economie. În 1946 și 1947, și din nou în mod independent, George B. Dantzig avea să dezvolte acest tip de sistem pentru aplicarea sa în probleme de planificare militară, iar mai târziu a descoperit o metodă elegantă și eficientă pentru a rezolva majoritatea problemelor de optimizare liniară, celebra *metodă simplex* (despre care vom vorbi, de asemenea, în capitolul 4).

EXEMPLUL 1

Să începem cu un exemplu simplu, academic. Să presupunem că avem două variabile, x_1 și x_2 , care pot lua valori reale non-negative, și că dorim să le maximizăm suma $C(x_1, x_2) = x_1 + x_2$. Pe lângă faptul că sunt non-negative ($x_1 \geq 0$, $x_2 \geq 0$), impunem următoarele restricții suplimentare asupra variabilelor: