Uvod u dinamičko programiranje

i primenu grafova u zaključivanju

Srđan Brkić

I izdanje

Beograd, 2024. godine

Srđan Brkić

UVOD U DINAMIČKO PROGRAMIRANJE
I PRIMENU GRAFOVA U ZAKLJUČIVANJU

Recenzenti
Dr Predrag Ivaniš
Dr Goran Marković

Odlukom Nastavno-naučnog veća Elektrotehničkog fakulteta br. 920/20 na 897. sednici održanoj 14.05.2024. ova knjiga je odobrena kao nastavni materijal - udžbenik na Elektrotehničkom fakultetu u Beogradu.

Izdavač
Akademska misao, Beograd

Dizajn naslovne strane
Boris Popović

Štampa
Akademska misao, Beograd

Tiraž
200 primeraka

ISBN 978-86-6200-011-8

Mesto i godina izdanja: Beograd, 2024.

Predgovor

Materijal predstavljen u ovom udžbeniku izlaže se u okviru kursa Algoritmi za dinamičku optimizaciju na četvrtoj godini osnovnih studija Odseka za Telekomunikcije, Elektrotehničkog fakulteta, Univerziteta u Beogradu. Takođe, udžbenik može poslužiti i studentima drugih odseka, kao i diplomiranim inženjerima elektrotehnike, koji žele da steknu ili prošire znanje o dinamičkom programiranju i metodama zaključivanja koji ga koriste.

Udžbenik je pisan sa željom da predznanje potrebno za razumevanje najvećeg broja koncepata opisanih u udžbeniku bude minimalno, a mišljenje autora je da osim osnovnih pojmova teorije grafova i teorije verovatnoće druga specijalna predznanja nisu nužna za praćenje izložene materije. Udžbenik predstavlja pokušaj autora da predstavi različite metode koje se koriste u oblasti nauke o podacima (eng. *data science*) i mašinskog učenja (eng. *machine learning*) iz ugla teoretičara informacija, sa idejom da bude koristan i studentima doktorskih studija koji se bave teorijom informacija i imaju potrebu da u svojim istraživanjima koriste dostignuća mašinskog učenja ili teorije ekspertskih sistema.

Autor bi posebno želeo da se zahvali recezentima dr Predragu Ivanišu, redovnom profesoru i dr Goranu Markoviću, vanrednom profesoru, na korisnim sugestijama i diskusijama.

Sadržaj

1 Optimizacioni algoritmi i kompleksnost

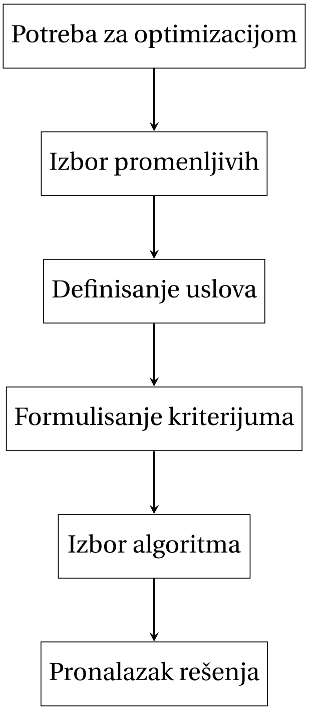
U ovom poglavlju biće objašnjeni osnovni koncepti matematičke optimizacije, sa posebnim osvrtom na diskretnu optimizaciju kojoj je posvećen ovaj udžbenik. Teorija kompleksnosti igra značajnu ulogu u rešavanju problema diskretne optimizacije, pa je deo poglavlja posvećen osnovama teorije komplesnosti sa nekoliko ilustrativnih primera. U poglavlju je takođe moguće pronaći i osnove dinamičke optimizacije sa prikazima nekoliko posebno značajnih algoritama, kojim se rešavaju pojedini problemi diskretne optimizacije.

1.1 Matematička optimizacija

Matematička optimizacija bi se grubo mogla nazvati selekcijom najboljeg elementa iz skupa elemenata pod zadatim uslovom, ali u širem (inženjerskom) smislu optimizacija bi uključivala i niz radnji koje se često nazivaju **domenskim operacijama** (slika 1.1) i svode se na pronalaženje problema koji se rešava optimizacijom (*potreba za optimizacijom*), *izbor promenljivih* koje se optimizuju, kao i *definisanje uslova* koje promenljive moraju da zadovolje. Broj promenljivih treba smanjiti na najmanji mogući broj iz razloga kompleksnosti. Uslovi predstavljaju fundamentalne relacije između promenljivih. Mogu predstavljati jednakosti ili nejednakosti, biti linearni ili nelinearni. Nakon definisanja uslova, pristupa se samoj matematičkoj optimizaciji u užem smislu. *Formulisanje kriterijuma* se odnosi na pronalaženje matematičkog opisa funkcije (metrike), koja se želi optimizovati. Uobičajno je da jedan problem može biti rešen različitim algoritmima, koji će se razlikovati prema nivoima računarske kompleksnosti potrebne za njihovo korišćenje.
Optimizacioni algoritmi se mogu podeliti na nekoliko načina koji će u nastavku biti prikazani.

- **Kontinualna** i **kombinaciona optimizacija** – kombinaciona optimizacija uključuje optimizaciju promenljivih iz nekog konačnog skupa, dok kontinualna optimizacija podrazumeva promenljive iz skupa realnih (kompleksnih) brojeva.

- **Uslovna** ili **bezuslovna optimizacija** – kada priroda problema koji se rešava dovodi do ograničenja vrednosti promenljivih, govori se o uslovnoj optimizaciji. Na



Slika 1.1: Dijagram matematičke optimizacije.

primer, ako se optimizuje diskretna raspodela nekog procesa, tada zbir svih promenljivih mora da bude jedinica, a same promenljive se nalaze u opsegu [0,1].

- **Deterministička** ili **stohastička optimizacija**. Nekada su podaci, koji su podložni optimizaciji, slučajni procesi, pa je cilj optimizacije zapravo izboriti se sa slučajnošću (najčešće usled neželjenog uticaja šuma, odnonsno merne nesigurnosti). Deterministička optimizacija pronalazi traženo rešenje kao tačku u predvidivoj višedimenzionoj funkciji.

U nastavku će, pre daljeg izlaganja, biti formalno definisani kontinualni i diskretni optimizacioni problemi [1].

Definicija 1.1.1:

Neka je data višedimenziona funkcija $f(\mathbf{x})$. Kontiualana optimizacija se definiše kao

$$
\begin{aligned}
&\underset{\mathbf{x}}{min} f(\mathbf{x}) \\
\text{uslovi: } &g_i(\mathbf{x}) \le 0,\ i = 1,2,\ldots,m \\
&h_j(\mathbf{x}) = 0,\ i = 1,2,\ldots,p
\end{aligned} \tag{1.1}
$$

Definicija 1.1.2:

Diskretna optimizacija se definiše kao uređena trojka (I, f, m) definisana u nastavku.

- *I je konačan skup ulaznih podataka.*
- *Za svaki podatak $\mathbf{x} \in I$, vrednost $f(\mathbf{x})$ predstavlja skup potencijalnih rešenja optimizacionog problema.*
- *Ulazni podatak $\mathbf{x}$ i potencijalno rešenje $y \in f(\mathbf{x})$ vezani su metrikom $m(\mathbf{x}, y)$.*

Optimalno rešenje y_{opt} zadovoljava uslov

$$m(\mathbf{x}, y_{opt}) = min\{m(\mathbf{x}, y) | y \in f(\mathbf{x})\}. \tag{1.2}$$

Po konvenciji, kontinualna optimizacija minimizuje vrednost funkcije, dok su funkcije uslova, koje promenljive moraju da zadovolje, manje ili jednake nuli. Kontinualna optimizacija potencijlano pretražuje beskonačan skup vrednosti i u opštem slučaju ne može se sa sigurnošću tvrditi da li optimalno rešenje postoji, niti da li je optimum dostignut u toku optimizacije. S druge strane, postavka diskretne optimizacije je takva da garantuje postojanje globalnog rešanja, što je fundamentalna razlika između dva postupka. Sirovom (eng. *brute force*) pretragom svih podataka iz skupa I sigurno se nalazi najbolje rešenje. Zbog toga bi se moglo pomisliti da su problemi diskretne optimizacije jednostavniji za rešiti. Međutim, sledeći primer putujućeg trgovca će biti iskorišćen da opovrgne prethodno zapažanje. Zadatak putujujućeg trgovca je da obiđe 15 različitih gradova tako da put koji pređe obilazeći gradove bude najmanji mogući. Trgovac poznaje međusobna rastojanja između gradova i može proveriti sve moguće putanje kojih ima 15! = 43.589.145.600. Ovaj broj je isuviše veliki da bi se u razumnom vremenu proverio na standarnom računaru, pa pronalazak najboljeg rešenja prostom pretragom nije verovatan. Sledi da pronalaženje optimalnog rešenja direktno zavisi od veličine skupa u kome se optimalno rešenje traži (zamisliti da trgovac treba da obiđe 30 gradova). U praksi su najčešći problemi koji ne mogu da se reše direktnom pretragom, već zahtevaju znatno sofisticiraniji pristup. Drugim rečima, kompleksnost algoritma kojim se rešava optimizacioni problem ima istaknutu ulogu u diskretnoj optimizaciji.

1.2 Kompleksnost algoritama

Kompleksnost optimizacionih algoritama se najčešće vezuje za vreme potrebno da se algortiam izvrši, kada se govori o **vremenskoj kompleksnosti** ili hardverskim resursima potrebnim za puštanje algoritma, kada se govori o **prostornoj kompleksnosti**. Prostorna kompleksnost algoritama najčešće se vezuje za memoriju koju je potrebno rezervisati u toku obavljanja operacija algoritma, što najčešće ne zavisi samo od algoritma, već i od hardverskog procesora na kome se pokreće algoritam. S druge strane, vremenska kompleksnost se meri brojem elementarnih računarskih operacija koje aloritam koristi i manje je zavisna od procesora na kome se one izvršavaju. Arhitektura procesora definiše način raspoređivanja operacija (neke operacije se izvršavaju serijski, a neke paralelno) ali ne menja njihov ukupan broj. Zbog toga se vremenska kompleksnost često smatra univerzalnijom metrikom kojom se opisuju i porede algoritmi i u nastavku ovog udžbenika, kada se drugačije ne naznači, kompleksnost algoritma će se odnositi isključivo na vremensku kompleksnost.

Prema dominantnoj paradigmi prisutnoj u informaciono-tehnološkoj industriji, inženjersko vreme je najvredniji kompanijski resurs, pa se u cilju smanjivanja vremena provedenog za rešavanje nekog problema često tolerišu i algoritamska rešenja, koja su kompleksnija od optimalno mogućih. Ubrzani razvoj poluprovodničkih komponenti, omogućio je konstrukciju bržih i efikasnijih računara, a samim tim i udobniji inženjerski rad iz kog je i proistekla navedena paradigma. Tako je moguće zamisliti veliki broj primena, u kojima korisniku nekog softverskog alata nije previše bitno da li se programski kod izvršava za 1ms ili 0,1s. Međutim, korišćenje neoptimalnog algoritma na duže staze može da bude problematičnije od povećanog inženjerskog vremena utrošenog da se pronađe optimalno rešenje. Naime, paralelno sa razvojem računara, povećava se i količina podataka koje ti računari treba da obrade – pri čemu količina podataka pokazuje brži rast od poznatog Murovog zakona vezanog za razvoj poluprovodničkih komponenti. Imajući to u vidu, dolazi se do zaključka da svaki neoptimalno rešen problem može prerasti u nerešen problem u doglednoj budućnosti. Zbog toga je od značaja takozvana **asimptotska**