The 9th Conference on Mathematics in Engineering: Theory and Applications
Novi Sad, May 31st–June 2nd 2024

DEKOMPOZICIONE METODE U DISTRIBUIRANOJ OPTIMIZACIJI BEZ OGRANIČENJA[1]

Tijana Ostojić [2] i Manojlo Vuković [3]

https://doi.org/10.24867/META.2024.17 Review article

Sažetak. Zbog sve većeg obima podataka u mašinskom učenju, kao i dubokom učenju, potreba za rešavanjem optimizacionih problema u distribuiranom okruženju kontinuirano raste, jer ovi podaci rezultiraju problemima sa mnogo promenljivih. Jedna od metoda koja se može koristiti i u distribuiranoj optimizaciji je dekompozicija promenljivih, koja je inicijalno korišćena u ranim 1960-im godinama. Ključna ideja za rešavanje navedenih velikih optimizacionih problema leži u razdvajanju skupa promenljivih na manje podskupove. Ovi podskupovi se zatim obrađuju nezavisno ili u koordinaciji, što olakšava rešavanje složenih problema. Stoga, metoda je posebno korisna u distribuiranim sistemima, gde se problem može podeliti na različite čvorove ili procesore. U okviru ovog preglednog rada biće dat prikaz osnovnih ideja primene dekompozicije u optimizaciji.

AMS klasifikacija (2020): 49M27, 49M29

Ključne reči: optimizacija, distribuirana optimizacije, dekompozicija

1. Uvod

Usled rastućih dimenzija i složenosti modernih skupova podataka, potreba za proučavanjem metoda za rešavanje optimizacionih problema sa velikim brojem varijabli postaje sve veća. Jedan od načina da se poboljša efikasnost rešavanja takvih problema je da se oni podele na više manjih problema, koji se mogu rešavati paralelno i nezavisno jedan od drugog smanjujući ukupno vreme potrebno za optimizaciju. Dakle, ovi problemi se mogu predstaviti kao optimizacija funkcija koje su predstavljene kao suma velikog broja pojedinačnih funkcija, gde svaka od njih može biti povezana sa određenim delom podataka ili zadatkom. Tako, rešavanje originalnog problema, koji bi se rešavao na jednom računaru, može da se distribuira na više računara koji zajedno dolaze do rešenja posmatranog optimizacionog problema. Mnogi aktuelni problemi u mašinskom učenju (kao npr. problemi konačnih suma $f(x) = \sum_{i=1}^{n} f_i(x)$), statistici, nauci

[1]Ovo istraživanje je podržano od strane Fakulteta tehničkih nauka Univerziteta u Novom Sadu kroz projekat "Unapređenje nastavnog procesa na engleskom jeziku u opštim disciplinama".

[2]Departman za opšte discipline u tehnici, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, e-mail: tijana.ostojic@uns.ac.rs

[3]Departman za opšte discipline u tehnici, Fakultet tehničkih nauka, Univerzitet u Novom Sadu, e-mail: manojlo.vukovic@uns.ac.rs

o podacima se mogu efikasnije rešiti korišćenjem algoritmima distribuirane optimizacije [1]. Dakle, potreba za distribuiranim okruženjem se ogleda u tome što su radni podaci isuviše veliki da bi se obrađivali i skladištili na jednom računaru, kao i činjenica da lokalna optimizacija često zahteva manje resursa i vremena nego globalna optimizacija. [11].

U okviru ovog preglednog rada posmatra se problem optimizacije bez ograničenja sledećeg oblika

$$\min_{x} f(x), \tag{1.1}$$

pri čemu će akcenat biti stavljen na dekompoziciju promenljivih kao jednog od pogodnih načina za rešavanje tog problema [3]. Problem (1.1) ima jedinstveno rešenje ako važi sledeća pretpostavka.

Pretpostavka 1.1. Funkcija $f : \mathbb{R}^n \to \mathbb{R}$ je konveksna i dva puta neprekidno diferencijabilna funkcija za koju postoje konstante $0 < \mu \leq L < \infty$ takve da za svako $x \in \mathbb{R}^n$ važi $\mu\mathbf{I} \preceq \nabla^2 f(x) \preceq L\mathbf{I}$.

U zavisnosti od osobina koje poseduje funkcija $f(x)$, u literaturi su predloženi razni metodi za rešavanje problema (1.1) [2, 4, 8, 9]. U ovom radu ćemo se koncentrisati na iterativni postupak linijskog pretraživanja koji je našao široku primenu u praksi [6, 8, 10]. Suština ovog postupka je da se dobije manja vrednost funkcije cilja u narednoj tački u odnosu na trenutnu. Iteracija se definiše na sledeći način

$$x_{k+1} = x_k + \alpha_k g_k. \tag{1.2}$$

Preciznije, x_{k+1} se računa tako da se prvo odredi opadajući pravac pretraživanja g_k[4], a potom dužina koraka α_k tako da se umanji vrednost funkcije cilja.

Dakle, aproksimacija rešenja originalnog problema se može dobiti korišćenjem sledećeg algoritma.

ALGORITAM 1.

za početnu tačku $x \in \mathbf{dom} f$

ponovi

S1 Nađi pravac pretraživanja g.

S2 Nađi dužinu koraka α,

S3 Ažuriraj $x = x + \alpha g$.

sve dok nije zadovoljen izlazni kriterijum.

Sledeća teorema daje uslove za konvergenciju niza $\{x_k\}$ koji je generisan sa (1.2).

[4] U literaturi su prisutni i razne metode koje ne pretpostavljaju da je pravac pretraživanja opadajući. U tom slučaju predložene su različite varijante nemonotonog linijskog pretraživanja (pogledati npr. [7].

Teorema 1.2. *[4] Neka je funkcija f zadovoljava pretpostavku 1.1. Neka je x_0 početna tačka i x_k dato sa (1.2), gde je g_k vektor dimenzije n i $\alpha_k \geq 0$ skalar. Neka je:*

1. *skup $S = \{x : f(x) \leq f(x_0)\}$ je ograničen,*
2. *vektori g_k zadovoljavaju dovoljan uslov pada*

$$-\frac{g_k^\top \nabla f(x_k)}{\|g_k\| \|\nabla f(x_k\|} \geq \epsilon > 0,$$

3. *za vektore g_k važi*

$$\|g_k\| \geq m \|\nabla f(x_k)\| \text{ za sve } k \ (m > 0),$$

i

$$\|g_k\| \leq M \text{ za sve } k,$$

4. *skalar α_k je izabran kao prvi element niza $1, \frac{1}{2}, \frac{1}{4}, \ldots$ koji zadovoljava dovoljan uslov pada*

$$f(x_k + \alpha g_k) \leq f(x_k) + \mu \alpha_k g_k^\top \nabla f(x_k),$$

gde je $0 < \mu < 1$.

Tada je

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0.$$

S obzirom na to da funkcija f zadovoljava pretpostavku 1.1, Teorema 1.2 garantuje konvergenciju niza $\{x_n\}$ ka rešenju problema (1.1). U ovom radu će se razmatrati primena Algoritma 1 u dekompoziciji.

Dekompozicija u optimizaciji predstavlja relativno staru ideju koja datira još u 1960-im godinama [5]. Motivacija za uvođenje ovih metoda leži u rešavanju problema veoma velikih dimenzija koji se u to vreme nisu mogli rešiti standarnim poznatim alatima. Danas se ona može koristiti u distribuiranoj optimizaciji. Pod uslovom da se promenljive u funkciji cilja $f(x)$ mogu podeliti, tj. $x = (x_1, x_2)$, problem (1.1) se može rešiti distribuirano. U sledećoj definiciji se uvodi pojam separabilnog problema optimizacije.

Definicija 1.3. Problem optimizacije je separabilan ako se može napisati na sledeći način

$$\min_{x=(x_1,x_2)} f_1(x_1) + f_2(x_2). \tag{1.3}$$

Primetimo da funkcija f_1 zavisi samo od x_1, a funkcija f_2 zavisi samo od x_2. Dakle, kako x_1 nije ni u kakvoj vezi sa x_2, problem (1.3) se može rešiti tako što se podeli na dva odvojena potproblema optimizacije:

$$\underbrace{\min f_1(x_1)}_{\text{Potproblem 1}} \quad \text{i} \quad \underbrace{\min f_2(x_2).}_{\text{Potroblem 2}}$$

Neka funkcije f_1 i f_2 zadovoljavaju Pretpostavku 1.1. Potproblem 1 i Potroblem 2 se mogu rešiti korišćenjem Algoritma 1 na dva različita računara (procesora) i samim tim skratiti vreme potrebno za rešavanje originalnog problema (1.3). Međutim, većina realnih problema u praksi nisu separabilni, ali se u nekim slučajevima mogu, korišćenjem dekompozicije, napraviti da to budu. U nastavku sledi definicija skoro separabilnog problema optimizacije, koji se, iako to nije, ipak može rešiti distribuirano korišćenjem dekompozicije.

Definicija 1.4. Problem optimizacije je skoro separabilan ako se može napisati na sledeći način

$$\begin{aligned} \min\ f(x) &= f_1(x_1, y) + f_2(x_2, y), \\ x &= (x_1, x_2, y). \end{aligned} \tag{1.4}$$

Primetimo da za fiksirano y, problem (1.4) postaje separabilan po x_1 i x_2, pa se pomenuta dva potproblema mogu rešiti totalno nezavisno. Kako y komplikuje problem, jer se javlja kao promenljiva u obe funkcije, ona se naziva *komplikovana promenljiva.* S druge strane, x_1 i x_2 se nazivaju *privatnim* ili *lokalnim promenljivima* koje se odnose na funkcije f_1 i f_2, respektivno, dok je y *javna promenljiva* koja vezuje ta dva potproblema.

2. Primarna dekompozicija

U nastavku će biti razmotreno rešavanje skoro separabilnog problema (1.4) korišćenjem primarne dekompozicije. Ideja u primarnoj dekompoziciji je da se za fiksirano y posmatraju sledeća dva potproblema

$$\underbrace{\min_{x_1} f_1(x_1, y)}_{\text{Potproblem 1'}} \quad \text{i} \quad \underbrace{\min_{x_2} f_2(x_2)}_{\text{Potroblem 2'}}.$$

Označimo sa $\Phi_1(y)$ i $\Phi_2(y)$ optimalne vrednosti Potproblema 1' i 2', respektivno[5]. Ono što se sada može primetiti jeste da se Potproblemi 1' i 2' mogu rešiti nezavisno (paralelno), korišćenjem Algoritma 1, a čak mogu biti i fizički odvojeni, tj. da se rešavaju na različitim računarima.

Lako je primetiti da je problem (1.4) ekvivalentan problemu

$$\min_y \Phi_1(y) + \Phi_2(y). \tag{2.1}$$

Odnosno, promenljive u (2.1) su komplikovane promenljive y originalnog problema, dok je funkcija cilja master problema zapravo suma optimalnih vrednosti potproblema. Ovaj deo rada zaključujemo sa algoritmom za rešavnje problema (1.4) pomoću primarne dekompozicije:

ALGORITAM 2.

ponovi

[5]Pretpostavljamo da je za funkcije f_1 i f_2 Pretpostavka 1.1 zadovoljena.

S1 Reši potprobleme (nezavisno).
Nađi x_1 koji minimizira $f_1(x_1, y)$ i gradijent $g_1 = \nabla\Phi_1(y)$.
Nađi x_2 koji minimizira $f_2(x_2, y)$ i gradijent $g_2 = \nabla\Phi_2(y)$.

S2 Ažuriraj komplikovanu promenljivu.
$y := y - \alpha_k(g_1 + g_2)$.

Primetimo da se u Algoritmu 2 koristi Algoritam 1 i za rešavanje potproblema i za rešavanje problema (2.1) jer se u koraku S1 računaju: 1) x_1 i x_2 koji minimiziraju potprobleme, 2) gradijenti $\nabla\Phi_1(y)$ i $\nabla\Phi_2(y)$ čiji se zbir dalje koristi kao pravac u koraku S2 kada se radi linijsko pretraživanje problema (2.1).

3. Dualna dekompozicija

Originalni skoro separabilni problem (1.4) se može zapisati i kao

$$\begin{aligned} &\min f(x) = f_1(x_1, y_1) + f_2(x_2, y_2) \\ &\text{pod uslovom } y_1 = y_2, \end{aligned} \tag{3.1}$$

gde su y_1, y_2 lokalne verzije komplikovane promenljive y. Primetimo da se Lagranžova funkcija dualnog problema (3.1)

$$L(x_1, y_1, x_2, y_2) = f_1(x_1, y_1) + f_2(x_2, y_2) + \nu^T(y_1 - y_2)$$

može zapisati na sledeći način

$$L(x_1, y_1, x_2, y_2) = \underbrace{f_1(x_1, y_1) + \nu^T y_1}_{\text{zavisi samo od } (x_1, y_1)} + \underbrace{f_2(x_2, y_2) - \nu^T y_2}_{\text{zavisi samo od } (x_2, y_2)}.$$

Dualni problem će biti separabilan po (x_1, y_1) i (x_2, y_2), tj. rešavanje dualnog problema može se svesti na rešavanje dva odvojena potproblema

$$\underbrace{g_1(\nu) = \inf_{x_1, y_1} f_1(x_1, y_1) + \nu^T y_1}_{\text{Potproblem 1''}} \quad \text{i} \quad \underbrace{g_2(\nu) = \inf_{x_2, y_2} f_2(x_2, y_2) - \nu^T y_2}_{\text{Potproblem 2''}}, \tag{3.2}$$

gde je dualni problem dat sa

$$\max g(\nu) = g_1(\nu) + g_2(\nu).$$

Analogno, kao i primarnoj dekompoziciji, rešavanje dualnih potproblema se može uraditi u potpunosti nezavisno. Prilagođavanjem Algoritma 1 može dobiti rešenje Potproblema 1" i 2", kao i rešenje problema (3.2). Algoritam za rešavanje problema (1.4) dualnom dekompozicijom je dat sa:

ALGORITAM 3.

ponovi

S1 Reši dualne potprobleme (nezavisno).
Nađi x_1, y_1 koji minimizira $f_1(x_1, y_1) + \nu^T y_1$.
Nađi x_2, y_2 koji minimizira $f_2(x_2, y_2) - \nu^T y_2$.

S2 Ažuriraj dualne promenljive.
$\nu := \nu - \alpha_k(y_2 - y_1)$.

Dakle, u koraku S1 se rešavaju dualni potproblemi i u koraku S2 zatim ažuriramo dualnu promenljivu koristeći gradijent funkcija g_1 i g_2.

4. Zaključak

U okviru ovog preglednog rada su izložene dve mogućnosti za rešavanje skoro separabilnih optimizacionih problema korišćenjem primarne i dualne dekompozicije. Mogućnost primene primarne i dualne dekompozicije na distribuiranu optimizaciju se ne zaustavlja samo na optimizaciji bez ograničenja, već se slične ideje mogu iskoristiti i na optimizaciju sa ograničenjima [3]. Ipak, autori su odlučili da se u ovom radu fokusiraju isključivo na primenu dekompozicionih metoda u optimizaciji bez ograničenja kako bi jasno i sažeto objasnili osnovne principe primarne i dualne dekompozicije. Takođe, bitno je napomenuti da uz odgovarajuće modifikacije navedenih algoritama i izbora pravca pretraživanja, dekompozicija se može uspešno primeniti na klasu semi-glathih funkcija. Korišćenje subgradijentnih metoda omogućava efikasno rešavanje optimizacionih problema i u slučajevima gde funkcije nisu glatke, tako da ovo daje nove mogućnosti za primenu dekompozicionih tehnika u širem spektru praktičnih problema.

Literatura

[1] S.P. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, vol. 3 of *Foundations and Trends in Machine Learning*, 2011.

[2] S.P. Boyd and L. Vandenberghe, *Convex optimization, Cambridge university press,* 2004.

[3] S.P. Boyd, L. Xiao, A. Mutapcic and J. Mattingley, *Notes for EE364B, Stanford University*, 2007.

[4] I. Griva, S. G. Nash and A. Sofer, *Linear and Nonlinear Optimization 2nd Edition. Society for Industrial and Applied Mathematics,* 2008.

[5] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, vol. 8 of *Operation research*, 1960.

[6] N. Krejić, and N. Krklec, *Line search methods with variable sample size for unconstrained optimization,* Journal of Computational and Applied Mathematics, 245, 213-231, 2013.

[7] N. Krejić, and N. Krklec, *Nonmonotone line search methods with variable sample size,* Numer Algor 68, 711–739, 2015.

[8] J. Nocedal and S. J. Wright, *Numerical optimization. New York, NY: Springer New York.* 1999.

[9] T. Ostojić, *Modifications of Newton-type methods for solving semi-smooth stochastic optimization problems.* PhD dissertation, Faculty of Sciences, Novi Sad, 2023.

[10] Z. J. Shi, *Convergence of line search methods for unconstrained optimization,* Applied Mathematics and Computation, 157(2), 393-405, 2004.

[11] Y. Zheng and Q. Liu, *A review of distributed optimization: Problems, models and algorithms* vol. 483 of *Neurocomputing*, 2022.