



UDK 519.6:004.42

NODIFFERENSIAL MASALALARNI YECHISHDA ZAMONAVIY SONLI USULLAR VA ULARNING DASTURIY REALIZATSIYASI

Saidnazarov Zoir Rahmon o'g'li

Surxondaryo viloyati Bandixon tumani 8-maktabda matematika o'qituvchi

saidnazarovzoir070797@gmail.com

Annotatsiya. Ushbu tezisdagi hosilasi mavjud bo'lmagan yoki ayrim nuqtalarda uzilish va "sinish"larga ega maqsad funksiyalari bilan bog'liq nodifferensial optimallashtirish masalalarini yechishning zamonaviy sonli yondashuvlari tahlil qilinadi. Subgradient, bundle, proksimal, gradient sampling hamda hosilasiz global qidiruv usullarining hisoblash xususiyatlari va qo'llanish chegaralari ko'rib chiqilgan. Shuningdek, mazkur algoritmlarning Python muhitida NumPy va SciPy kutubxonalari yordamida dasturiy realizatsiyasi, algoritmlar tanlashda aniqlik, barqarorlik va hisoblash xarajatlari o'rtasidagi muvozanat masalalari yoritilgan. Tahlil natijalari nodifferensial masalalarda bitta universal algoritmlar mavjud emasligini, masala strukturasi va o'lchamiga mos gibrid yondashuvlar amaliy jihatdan samaraliroq ekanini ko'rsatadi.

Kalit so'zlar: nodifferensial optimallashtirish, sonli usullar, subgradient usuli, bundle usuli, proksimal algoritmlar, gradient sampling, hosilasiz optimallashtirish, SciPy, Python.

Аннотация. В тезисе рассматриваются современные численные подходы к решению задач недифференцируемой оптимизации, в которых целевая функция не имеет производной либо содержит точки излома и разрывы производной. Анализируются субградиентные, пакетные (bundle), проксимальные методы, методы выборки градиента и безградиентного глобального поиска. Отдельное внимание уделено программной реализации алгоритмов в среде Python с использованием библиотек NumPy и SciPy, а также выбору метода с учетом точности, устойчивости и вычислительных затрат. Показано, что для недифференцируемых задач наиболее рационален структурно-ориентированный выбор алгоритма и сочетание локальных и глобальных процедур.

Ключевые слова: недифференцируемая оптимизация, численные методы, субградиентный метод, bundle-метод, проксимальный алгоритм, выборка градиента, безградиентная оптимизация, SciPy, Python.



Abstract. This thesis examines modern numerical approaches to nonsmooth optimization problems in which the objective function is not differentiable or contains kinks and discontinuities in its derivative. Subgradient, bundle, proximal, gradient-sampling, and derivative-free global-search methods are considered with emphasis on their computational properties and practical limitations. The software implementation of these algorithms in Python using NumPy and SciPy is also discussed, together with the trade-off between accuracy, stability, and computational cost. The analysis indicates that there is no universal solver for nonsmooth problems; in practice, structure-aware and hybrid strategies are often more effective.

Keywords: nonsmooth optimization, numerical methods, subgradient method, bundle method, proximal algorithm, gradient sampling, derivative-free optimization, SciPy, Python.

KIRISH

Ko'plab amaliy optimallashtirish masalalarida maqsad funksiyasi klassik ma'noda silliq bo'lmaydi. Absolyut qiymat, maksimum yoki minimum operatorlari, bo'lakli chiziqli bog'lanishlar, L1-regulyarizatsiya, kontakt va mexanika modellari, transport va logistika xarajatlari, mashinaviy o'qitishdagi ayrim yo'qotish funksiyalari bunga tipik misoldir. Bunday holatlarda Nyuton yoki oddiy gradient tushish usullarini bevosita qo'llash nazariy jihatdan asoslanmagan yoki amalda beqaror natija berishi mumkin. Shu sababli nodifferensial optimallashtirish alohida sonli usullar sinfini talab etadi (Gaudioso, Giallombardo, & Miglionico, 2022).

Nodifferensial optimallashtirishning dolzarbligi ikki omil bilan kuchaymoqda. Birinchidan, katta hajmli ma'lumotlar bilan ishlaydigan modellar ko'pincha siyrak yechim olish uchun L1-tipidagi regulyarizatsiyadan foydalanadi. Ikkinchidan, muhandislik va iqtisodiy modellarni real sharoitga yaqinlashtirish natijasida maqsad funksiyasida cheklovlar, jarima hadlari va bo'lakli ifodalar paydo bo'ladi. Demak, algoritim nafaqat minimumni topishi, balki hosila mavjud bo'lmagan nuqtalarda ham barqaror ishlashi kerak.

Tezisning maqsadi nodifferensial masalalarni yechishda qo'llanilayotgan asosiy zamonaviy sonli usullarni tizimlashtirish, ularning kuchli va zaif tomonlarini qiyoslash hamda dasturiy realizatsiya uchun amaliy tavsiyalar berishdan iborat.

ASOSIY QISM

Nodifferensial masalaning umumiy ko'rinishi $\min f(x)$, $x \in X$ tarzida yoziladi. Bu yerda $f(x)$ funksiyasi uzluksiz bo'lishi mumkin, biroq barcha nuqtalarda klassik gradientga ega bo'lishi shart emas. Konveks holatda gradient o'rniga subgradient tushunchasi qo'llanadi. g vektor f funksiyaning x nuqtadagi subgradienti bo'lishi uchun barcha y nuqtalarda $f(y) \geq f(x) + g^T(y - x)$



x) tengsizlik bajarilishi kerak. Mazkur yondashuv hosila mavjud bo'lmagan nuqtalarda ham yo'nalish haqida foydali ma'lumot beradi.

Subgradient usuli algoritmik jihatdan sodda: $x(k+1)=x(k)-\alpha(k)g(k)$, bunda $\alpha(k)$ qadam uzunligi. Uning asosiy ustunligi xotira talabi pastligi va katta o'lchamli masalalarga moslashuvchanligidir. Biroq qadamni noto'g'ri tanlash tebranish, sekin yaqinlashish yoki minimum atrofida uzoq yurishga olib keladi. Shu sababli amaliy dasturda kamayuvchi qadam, adaptiv qadam yoki Polyak tipidagi strategiyalar qo'llanadi.

Bundle usullarida esa faqat joriy subgradient emas, balki oldingi iteratsiyalarda olingan funksional qiymatlar va subgradientlar "paketi" saqlanadi. Shu ma'lumotlar asosida funksiyaning bo'lakli chiziqli modeli tuziladi va yangi qidiruv nuqtasi aniqlanadi. Bu usullar subgradient usuliga nisbatan ko'proq xotira va hisoblash talab qiladi, ammo nodifferensial konveks masalalarda barqarorroq va tezroq yaqinlashishi mumkin. Zamonaviy adabiyotlarda bundle metodlari nonsmooth optimizationning asosiy algoritmik yo'nalishlaridan biri sifatida qaraladi (Bagirov et al., 2020; Mäkelä, 2002).

Proksimal usullar maqsad funksiyasini ikki qismga ajratish mumkin bo'lgan holatlarda ayniqsa samarali: $F(x)=f(x)+g(x)$, bunda f silliq, g esa nodifferensial, lekin uning proksimal operatori hisoblanishi oson. Masalan, L1-regulyarizatsiya uchun soft-thresholding operatori qo'llanadi. Proksimal gradient, FISTA va ADMM kabi yondashuvlarning ommalashishi aynan shu strukturadan unumli foydalanishi bilan bog'liq. Ular tasvirlarni qayta ishlash, siyrak regressiya va mashinaviy o'qitish masalalarida katta amaliy ahamiyatga ega.

Gradient sampling usuli lokal Lipschitz funksiyalar uchun boshqa tamoyilga asoslanadi: joriy nuqta atrofidan bir nechta yaqin nuqtalar tanlanadi, ularda gradientlar hisoblanadi va ularning konveks qobig'i orqali umumlashtirilgan tushish yo'nalishi quriladi. Ushbu yondashuv silliq bo'lmagan, biroq deyarli hamma joyda differensiallanuvchi funksiyalarda foydali. Kamchiligi shundaki, har bir iteratsiyada bir nechta gradient baholash talab qilinadi, bu esa katta o'lchamli masalalarda hisoblash narxini oshiradi.

Ayrim masalalarda funksiya qiymatini hisoblash mumkin, ammo gradient yoki subgradientni olish qiyin. Bunday sharoitda hosilasiz usullar, jumladan Nelder–Mead, Powell, pattern search va evolyutsion algoritmlar qo'llanadi. SciPy kutubxonasidagi differential_evolution algoritmi gradientdan foydalanmaydigan stoxastik global optimallashtirish usuli bo'lib, katta qidiruv sohasini ko'rib chiqishi mumkin; buning evaziga funksiya hisoblashlar soni odatda gradientga asoslangan usullarga nisbatan ko'proq bo'ladi (SciPy Developers, 2026).



1-jadval. Nodifferensial masalalar uchun sonli usullarning qiyosiy tavsifi

Usul	Asosiy afzallik	Cheklov	Mos masalalar
Subgradient	Sodda, xotira talabi kichik	Yaqinlashishi sekin	Katta o'lchamli konveks masalalar
Bundle	Barqaror lokal model	Xotira va ichki QP xarajati	Konveks va ayrim nokonveks NSO
Proksimal	Strukturadan samarali foydalanadi	Prox-operator mavjud bo'lishi kerak	L1, siyrak va kompozit modellar
Gradient sampling	Nonsmooth lokal geometriyani hisobga oladi	Ko'p gradient baholash	Lokal Lipschitz funksiyalar
Differential evolution	Gradient talab qilmaydi, global qidiruv	Ko'p funksiya hisoblash	Murakkab, multimodal, black-box masalalar

Dasturiy realizatsiyada algoritmnı matematik xususiyatlarga mos tanlash muhim. Python muhiti ochiq kodli va qayta ishlab bo'ladigan hisoblash ekotizimini beradi. NumPy vektor va matritsa amallarini bajaradi, SciPy esa mahalliy va global optimallashtirish algoritmlarini taqdim etadi. Masalan, maqsad funksiya faqat qiymat qaytarsa va o'zgaruvchilar chegaralangan bo'lsa, `scipy.optimize.differential_evolution` yordamida global qidiruv amalga oshirilishi mumkin. Agar masala konveks va maxsus proksimal strukturaga ega bo'lsa, algoritmnı NumPy asosida alohida yozish ko'pincha maqsadga muvofiq bo'ladi. Katta o'lchamli modellar uchun vektorlashtirish, siyrak matritsalar va parallel hisoblash hisoblash vaqtini sezilarli qisqartiradi.

Masalan, $f(x)=|x_1|+2|x_2|+(x_1+x_2-1)^2$ kabi funksiya $x_1=0$ va $x_2=0$ chiziqlarida klassik ma'noda differensiallanuvchi emas. Bunday funksiya uchun oddiy markaziy ayirmalar yordamida gradientni taxminlash nodifferensial nuqta yaqinida keskin o'zgaruvchan yo'nalish hosil qilishi mumkin. Proksimal yoki subgradient yondashuv esa absolyut qiymat hadlarini alohida hisobga olib, iteratsion jarayonni nazariy jihatdan to'g'ri quradi. Bu misol algoritm tanlashda "qaysi dastur tezroq ishlaydi?" degan savoldan oldin "funksiya qanday strukturaga ega?" degan savol qo'yilishi zarurligini ko'rsatadi.



Zamonaviy dasturiy yechimlar uchun gibridd sxema ham istiqbolli hisoblanadi. Dastlab global hosilasiz algoritmlar bilan qidiruv sohasi toraytiriladi, soʻng topilgan nuqta atrofiga proksimal, bundle yoki boshqa lokal algoritmlar bilan aniqlik oshiriladi. Bunday yondashuv global algoritmlarning keng qidiruv imkoniyati bilan strukturaviy lokal metodlarning tez yaqinlashishini birlashtiradi. Biroq gibridd algoritmlar ham hisoblash budjeti, toʻxtash mezonlari, boshlangʻich nuqta va parametrlar sezgirlik boʻyicha sinovdan oʻtkazilishi kerak.

Sonli algoritmlarning sifatini faqat yakuniy funksional qiymat bilan baholash yetarli emas. Amaliy dasturda iteratsiyalar soni, funksiya va subgradient hisoblashlar soni, protsessor vaqti, xotira sarfi, boshlangʻich nuqtaga sezgirlik va natijaning takrorlanuvchanligi ham qayd etilishi lozim. Stoxastik algoritmlarda random seed qiymatini belgilash va bir nechta mustaqil ishga tushirishlar boʻyicha natijalarni taqqoslash ilmiy xulosaning ishonchliligini oshiradi.

XULOSA

Nodifferensial masalalarni sonli yechish klassik silliq optimallashtirishga nisbatan murakkabroq, chunki hosila mavjud boʻlmagan nuqtalarda odatiy gradient apparati yetarli emas. Subgradient usuli soddaligi va katta oʻlchamli masalalarga mosligi bilan ajralib turadi; bundle metodlari oldingi iteratsiyalar haqidagi maʼlumotni saqlab, barqarorroq model hosil qiladi; proksimal usullar esa kompozit va siyrak strukturalarda eng samarali vositalardan biridir. Gradient sampling lokal nonsilliqlikni hisobga olsa, hosilasiz evolyutsion usullar murakkab va black-box funksiyalarda foydali global qidiruv imkonini beradi.

Dasturiy realizatsiyada usul tanlash funksiyaning konveksligi, oʻlchami, subgradient yoki proksimal operator mavjudligi, hisoblash narxi va talab etiladigan aniqlikka bogʻliq. Shu sababli amaliy masalalarda universal algoritmlar izlashdan koʻra, masala strukturasi asosida metod tanlash va zarur hollarda global hamda lokal usullarni gibridd shaklda qoʻllash maqsadga muvofiq. Python, NumPy va SciPy kabi ochiq dasturiy vositalar bunday algoritmlarni tez prototiplash, solishtirish va takrorlanuvchi hisoblash tajribalarini tashkil etish uchun qulay muhit yaratadi.

FOYDALANILGAN ADABIYOTLAR

1. Bagirov, A. M., Gaudio, M., Karmitsa, N., Mäkelä, M. M., & Taheri, S. (Eds.). (2020). Numerical nonsmooth optimization: State of the art algorithms. Springer. <https://doi.org/10.1007/978-3-030-34910-3>
2. Beck, A. (2017). First-order methods in optimization. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611974997>
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1), 1–122. <https://doi.org/10.1561/22000000016>



4. Gaudioso, M., Giallombardo, G., & Miglionico, G. (2022). Essentials of numerical nonsmooth optimization. *Annals of Operations Research*, 314, 213–253. <https://doi.org/10.1007/s10479-021-04498-y>
5. Mäkelä, M. M. (2002). Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17(1), 1–29. <https://doi.org/10.1080/10556780290027828>
6. Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers.
7. Parikh, N., & Boyd, S. (2014). Proximal algorithms. *Foundations and Trends in Optimization*, 1(3), 127–239. <https://doi.org/10.1561/24000000003>
8. SciPy Developers. (2026). *scipy.optimize.differential_evolution* — SciPy reference guide. SciPy. https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html