Liet. matem. rink, **45**, *spec. nr.*, 2005, 404–408

Diskrečių resursų priskyrimas vartotojų grupėms

Vytautas TIEŠIS (MII, VPU)
el. paštas: tiesis@ktl.mii.lt

1. Įvadas

Nemažos dalies pseudosveikaskaitinio programavimo uždavinių sudėtingumas yra aukštesnis nei polinominis. Toks yra ir straipsnyje nagrinėjamas resursų paskirstymo uždavinys su netiesine pelno funkcija ir specifiniais ribojimais. Panašūs paprastesni uždaviniai yra dar Belmano spręsti dinaminio programavimo metodais [1]. Sudėtingesniems ir ypač didesnės apimties uždaviniams sunku taikyti tikslius metodus, nes operacijų kiekis auga eksponentiškai. Todėl taikyti apytiksliai matematinio programavimo metodai [2], [3], [4], [5]. Apytikslių algoritmų taikymą pateisina tai, kad ir naudojami modeliai paprastai yra apytiksliai.

Šiame straipsnyje pratęsiami ankstesni autoriaus ir kolegų darbai [6], [7] – pasiūlyti apytiksliai greiti algoritmai netiesiniam uždaviniui su specifiniais ribojimais: vienos rūšies resursai gali būti priskirti tik vienai vartotojų grupei, o kiekviena vartotojų grupė gali naudoti visų rūšių resursus. Rasti algoritmų operacijų skaičiaus įverčiai, eksperimentiškai pagrįstas algoritmo modifikacijos pranašumas. Sukurti algoritmai priklauso diskretinių gradientinių algoritmų klasei [2], [3], [4] ir yra žinomų algoritmų išvystymas, panaudojant specifines uždavinio savybes.

2. Uždavinio formulavimas

Aprašykime pseudosveikaskaitinio optimizavimo uždavinį su realia netiesine tikslo funkcija, kurį galima interpretuoti kaip diskrečių nevienarūšių resursų paskirstymo vartotojų grupėms optimizavimą. Tarkim, kad turime m rūšių resursų, kiekvienos rūšies po b_i vienetų. Tuos resursus reikia paskirstyti tarp n vartotojų, paskirstymą aprašo matrica $x = [x_{ij}],\ i = \overline{1, m},\ \ j = \overline{1, n}$, kur $x_{ij} \in \{0, 1, ..., b_i\}$ yra i-tos rūšies resursų kiekis, paskirtas j-tajam vartotojui. Visi j-tajam vartotojui paskirti resursai aprašomi vektoriumi $x_j = (x_{ij},\ i = \overline{1, m})$ ir j-tasis vartotojas padaro $\varphi_j(x_j)$ nuostolių. Bendri nuostoliai $F(x) = \sum_{j=1}^{n} \varphi_j(x_j)$ yra $n \times m$ sveikaskaitinių kintamųjų reali funkcija $F(x)$: $I^{n\times m} \to R$, adityvi vektorinių kintamųjų $x_j,\ j = \overline{1, n}$ atžvilgiu.

Vienos rūšies visi resursai gali būti priskirti vienai vartotojų grupei, ir tas vartotojų suskaidymas į grupes nebūtinai turi būti vienodas skirtingoms resursų rūšims. Tarkim, kad kiekvienam i vartotojų aibė $J = \{1, ..., n\}$ yra suskaidyta į K_i nepersikertančių grupių $A_{ik} \subset J,\ k = \overline{1, K_i},\ A_{ik} \cap A_{il} = \phi,\ k \neq l$.

Turime minimalių nuostolių paieškos uždavinį:

$$F(x) = \sum_{j=1}^{n} \varphi_j(x_j) \to \min_{x \in D}. \tag{1}$$

Kintamųjų x_{ij} sritis D apibrėžiama taip:

$$x_{ij} \in \{0, 1, ..., b_i\}, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \leqslant b_i, \quad i = \overline{1, m}, \ \ b_i > 0, \tag{3}$$

$$\sum_{j \in A_{il}} x_{ij} \cdot \sum_{s \in A_{ik}} x_{is} = 0 \quad \forall i, l, k\colon \ l \neq k. \tag{4}$$

Uždavinys (1)–(4) bendru atveju, taip pat ir dažnu specialiu atveju, yra sudėtingas, nesprendžiamas per polinominį laiką. Uždavinio sudėtingumą ir taikytinus algoritmus apsprendžia funkcijų $\varphi_j(x_j)$ ir srities D savybės.

Efektyvius sprendimo algoritmus sukurti mums padės kai kuriems taikymams būdinga tikslo funkcijos savybė: papildomų resurso vienetų paskyrimas vartotojui duoda vis mažesnį nuostolių sumažėjimą – papildomi resursai vis mažiau efektyviai naudojami. Nagrinėsime atvejį, kai tikslo funkcijos diskretinis kairysis gradientas [4]

$$\nabla_{ij}^{-} F(x + e_{ij}) \equiv F(x) - F(x + e_{ij}) = \varphi_j(x_j) - \varphi_j(x_j + e_i) \equiv \nabla_i \varphi_j(x_j) \tag{5}$$

yra nedidėjanti funkcija. Čia vektorius e_{ij} turi ij-tąją vienetinę koordinatę, atitinkamai e_i – i-tąją vienetinę koordinatę, kitos koordinatės nulinės. Tokios funkcijos vadinamos koordinatiškai iškiliomis. Laikysime, kad funkcijos $\varphi_j(x_j)$ irgi yra nedidėjančios.

3. Sprendimo algoritmai

Sveikaskaitinio optimizavimo uždaviniams spręsti sėkmingai naudojami diskretinio gradiento (greedy) metodai. Metode kiekvieno žingsnio metu paskiriama viena resurso porcija, duodanti didžiausią nuostolių funkcijos sumažėjimą.

Hierarchinis algoritmas. Uždaviniui (1)–(4) šį metodą skaidysime į du hierarchinius lygius. Aukštesnio lygio sprendimo algoritmas išrenka didžiausią naudą duodantį vienos rūšies visų resursų paskyrimą vienai vartotojų grupei. Tas paskyrimo naudas skaičiuoja žemesnio lygio algoritmas, kuris paskirsto vienos rūšies resursus grupės viduje, tai yra randa dalinio uždavinio (1)–(3) su $m = 1$ sprendinį. Aukštesnio lygio algoritmas kiekviename žingsnyje naudoja žemesnio lygmens algoritmą tiek kartų kiek liko galimų paskyrimų grupėms.

Žemesnio lygmens algoritmas vartotojų grupei nuosekliai skirstys po vieną vienos rūšies resurso vienetą – pirmiausia didžiausios naudos paskyrimai.

Kai $m = 1$ daliniams uždavinio (1)–(3) atvejams eilė autorių įrodė, kad toks nuoseklaus skirstymo algoritmo sprendinys yra globalus optimumas. Tam tikrais aspektais bendresnio uždavinio sprendinio optimalumą įrodė Kovaliovas [4], remdamasis tikslo funkcijos separabilumu ir koordinatiniu iškilumu.

Formalizuokime aukštesnio lygmens uždavinio algoritmą.

Pradedame nuo $x^0=0,\ \beta^0=(1,\dots,1),\ \psi^o=(\psi_1,\dots,\psi_n)=(\varphi_1(x_1^0),\dots,\varphi_n(x_n^0))$. Kiekviename c-tame žingsnyje randame kurios nors rūšies resursų optimalų paskyrimą vienai vartotojų grupei:

$$(l,k)=\arg\max_{(ir):\ \beta_i^c>0,\ 1\leqslant r\leqslant K_i}\bigl\{F(x^c)-F(x^c+z^{cir})\bigr\}. \tag{6}$$

Iš žemesnio lygmens grąžinami $x^{c+1}=x^c+z^{clk},\ \beta^{c+1}=\beta^c-e_l,\ \psi^{c+1}$; čia z^{cir} yra žemesnio lygmens sprendinys, gautas paskirstant i-tos rūšies resursus A_{ir} grupės viduje nuoseklaus skirstymo algoritmu.

Žemesnio lygmens uždaviniai aprašomi taip:

$$z^{cir}=\arg\min_{z\in D_{ir}}F(x^c+z),\quad \text{čia}\ D_{ir}=\Biggl\{z\colon \sum_{j\in A_{ir}}z_{ij}\leqslant b_i;\ (z_{lj}=0,\ j\notin A_{ir}\vee l\neq i)\Biggr\}.$$

Formalizuodami žemesnio lygmens algoritmą pasinaudosime tikslo funkcijos pokyčio separabilumu (5) ir tuo, kad priskyrus vieną resurso vienetą j-tam vartotojui, pakinta tik to vartotojo nuostoliai – vėlesniems palyginimams naudosime išsaugotą nuostolių vektorių, taip mažindami nuostolių skaičiavimų kiekį.

Skirstymą vartotojų grupei A_{ik} pradedame nuo kintamųjų $y^0=x^c,\ p^0=b_i,\ w^o=(\psi_1,\dots,\psi_n)=(\varphi_1(y_1^0),\dots,\varphi_n(y_n^0)),\ s=0$, perduotų iš aukštesnio lygio algoritmo. Pradinis išlošimas $\Delta=0$. Paskaičiuojame galimų sumažėjusių nuostolių vektorių $\gamma^o=(\gamma_j)=(\varphi_j(y_j^0+e_{ij})),\ j\in A_{ik}$.

Kiekviename s-tame žingsnyje randame optimalų paskyrimą:

$$l=\arg\max_j\bigl\{\nabla_i\varphi_j(y_j^s)=\psi_j-\gamma_j\bigr\}. \tag{7}$$

Atliekame perskaičiavimus $y^{s+1}=y^s+e_{il},\ p^{s+1}=p^s-1,\ \Delta=\Delta+\psi_l-\gamma_l,\ \psi_l=\gamma_l$, jei $p^{s+1}\neq 0$ tai $\gamma_l=\varphi_l(y_l^{s+1}+e_{il})$.

Skaičiavimus tęsiame tol, kol yra ką paskirti, tai yra kol taps $p^{s+1}=0$, arba $\nabla_i\varphi_l(y_l^{s+1})=0$. Tada y^{s+1} bus tikslus dalinio uždavinio sprendinys x^c+z^{cir}.

Algoritmo sudėtingumas. Žemesnio lygio algoritmas turės ne daugiau b_i žingsnių ir $\nabla\varphi$ bus skaičiuojamas ne daugiau $b_i|A_{ik}|$ kartų; vieno vartotojo nuostoliai φ skaičiuojami tik skaičiuojant vektorių γ, tai yra ne daugiau $b_i-1+|A_{ik}|$ kartų. Jei $\nabla_i\varphi_j(x_j),\ j\in A_{ik}$ yra griežtai mažėjančios funkcijos, tai skaičiavimų kiekių įverčiai yra tikslūs.

Atliekant perrinkimą (6) vienai resursų rūšiai dalinis uždavinys sprendžiamas tiek kartų, kiek yra grupių – K_i. Taigi, jei visi b_i lygūs b ir K_i lygūs K, tai vieno vartotojo nuostoliai φ skaičiuojami $\sum^{K_i}(|A_{ik}|+b_i-1)=n+K(b-1)$ kartų. Vienos rūšies grupių perrinkimas atliekamas $m(m+1)/2$ kartų, todėl turime teiginį:

TEIGINYS. *Bendras vieno vartotojo nuostolių funkcijos φ skaičiavimų kiekis hierarchiniu algoritmu yra nedidesnis nei $(n+Kb-K)m(m+1)/2$ ir lygus šiam dydžiui tada, kai diskretaus gradiento funkcijos $\nabla_i\varphi_j(x_j),\ j\in A_{ik}$ yra griežtai mažėjančios x_j atžvilgiu srityje D.*

Šakų ir ribų metodas. Hierarchinį algoritmą galima pagreitinti, tai yra sumažinti nuostolių skaičiavimų kiekį, nekeičiant algoritmo tikslumo. Šaka laikysime paskirstymą vienai vartotojų grupei. Riba – išlošimas Δ pasiektas ankstesnėse grupėse žingsnio (6) viduje. Spręsdami dalinį uždavinį (7) A_{ir} grupei, kiekviename nuoseklaus skirstymo algoritmo s-tame žingsnyje nuostolius sumažiname dydžiu $\nabla_i \varphi_j(x_j^s)$. Trumpumo dėlei pažymėkime šį dydį ∇_{ir}^s. Iš koordinatinio iškilumo išplaukia, kad su kiekvienu s žingsniu skirtumas ∇_{ir}^s vis mažės. Todėl po dalinio uždavinio sprendimo s-tojo žingsnio turime įvertį $F(x^c + z^{cir}) \geqslant F(x^c) - \sum_{h=0}^{s} \nabla_{ir}^h - (b_i - s - 1)\nabla_{ir}^s \equiv \omega_{ir}$. Kiekviename aukštesnio lygio algoritmo žingsnyje nutraukiame dalinio uždavinio grupei A_{ir} sprendimą, jei turime perspektyvesnę grupę A_{LK}, tai yra, jei anksčiau pasiektas rekordas $F(x^c + z^{cLK}) \leqslant \omega_{ir}$.

4. Pavyzdys ir eksperimentinis tyrimas

Aprašysime paprastą nuostolių funkcijos modelį. Tarkim, kad yra žinoma j-tojo taikinio sunaikinimo tikimybė a_{ij} $(i = \overline{1,m},\ j = \overline{1,n})$ su i-tos rūšies gynybos priemonės vienetu, o taip pat yra žinomas subjektyvus j-tojo taikinio svoris, kitaip tariant, galimas nuostolis $G_j(j = \overline{1,n})$, kurį gali sukelti j-tasis taikinys. Laikome, kad taikinio išlikimo faktai po kiekvienos priemonės vieneto panaudojimo yra nepriklausomi įvykiai. Tada vidutiniai galimi nuostoliai $\varphi_j(x_j)$ dėl j-tojo taikinio yra $\varphi_j(x_j) = G_j \prod_{i=1}^{m} (1 - a_{ij})^{x_{ij}}$.

1 lentelėje pateikiame vieno vartotojo nuostolių funkcijos φ skaičiavimų kiekio vidurkius IF, gautus iš 100 atsitiktinai sugeneruotų uždavinių.

5. Išvados

Pasiūlytas greitas hierarchinis gradientinis algoritmas nevienalyčiams diskretiems resursams skirstyti vartotojų grupėms, kai tikslo funkcijos yra netiesinės ir koordinatiškai iškilios. Gavome tikslų algoritmo sudėtingumo polinominį įvertį. Eksperimentai rodo, kad algoritmo modifikacija dar labiau sumažina skaičiavimų kiekį, ypač didelės apimties uždaviniams. Santykinės paklaidos kinta nuo 1 iki keliolikos procentų augant uždavinio apimčiai.

1 lentelė. Vieno vartotojo nuostolių funkcijos skaičiavimų kiekiai

K	2	2	2	3	3	3	5	10
m	2	2	4	4	6	6	30	30
n	5	5	5	10	10	10	30	50
b	2	4	4	4	4	8	30	30
Hierarchinio IF_H	27	39	130	120	462	714	83700	162750
Šakų ir ribų IF_S	20	30	97	160	329	481	37572	54815
$100IF_S/IF_H\%$	75	79	75	73	71	67	45	34

Literatūra

1. R.E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, USA (1957).
2. М.М. Ковалев, *Матройды в дискретной оптимизации*, Издательство „Университетское“, Минск (1987).
3. E. Girlich, M.M. Kovalev, D.M. Vasilkov, Greedy sets and related problems, *European J. of Operational Research*, **101**, 74–80 (1997).
4. М.М. Ковалев, Метод частичных порядков, *Докл. АН БССР*, **24**(2), 113–116 (1980).
5. A. Andersson, P. Carlsson, F. Ygge, Resource allocation with Wobbly functions, *Computational Optimization and Applications*, **23**, 171–200 (2002).
6. В. Тешис, Градиентные алгоритмы распределения порционного ресурса, *Optimalių sprendimų teorija*, **14**, 139–159 (1990).
7. В. Тешис, В. Шалтянис, Ф. Юшка, О задаче распределения порционного ресурса, *Optimalių sprendimų teorija*, **12**, 100–113 (1987).

SUMMARY

V. Tiešis. Discrete resource allocation to groups of customers

The paper deals with the non-homogeny discrete resource allocation to groups of customers with non-linear profit functions. The fast hierarchical greedy algorithms are presented and investigated. The exact upper bound was found for complexity of the algorithm. The experiments prove the advantage of the modification of the algorithm.

Keywords: discrete optimisation, greedy algorithms, resource allocation.