ISSN 1648-8776
JAUNŲJŲ MOKSLININKŲ DARBAI. Nr. 1 (45). 2016 Doi: 10.21277/jmd.v1i45.42

DVIEJŲ ETAPŲ STOCHASTINIO TIESINIO PROGRAMAVIMO UŽDAVINIO AGREGAVIMO TYRIMAS

Ana Ušpurienė[1], Leonidas Sakalauskas[2]
[1]Vilniaus universiteto Matematikos ir informatikos institutas, [2]Šiaulių universitetas

Įvadas

Dviejų arba kelių etapų stochastinio tiesinio programavimo (STP) uždaviniai yra klasikinio tiesinio programavimo apibendrinimas, kai uždavinio parametrai yra atsitiktiniai kintamieji. Šiame darbe nagrinėjamas dviejų etapų STP uždavinys, kai antro etapo kintamieji priklauso nuo atsitiktinio parametro. Modeliuojant atsitiktinį procesą, generuojami atsitiktiniai scenarijai pagal normalųjį (Gauso) dėsnį. Sprendžiant šį uždavinį, naudojamas modifikuotas L pavidalo (L-shaped) algoritmas. Norint pasiekti tikslesnį rezultatą, optimizavimo proceso metu generuojamų scenarijų skaičius turi būti pakankamai didelis. Kai uždavinio kintamųjų skaičius yra didelis, optimizavimas užima daug laiko ir sunaudoja daug resursų. Siekiant sumažinti optimizavimo proceso iteracijų skaičių, naudojamų resursų kiekį ir optimalaus sprendinio gavimo skaičiavimo laiką, darbe nagrinėjamas scenarijų agregavimo metodas, pritaikytas sprendžiant dviejų etapų STP uždavinį.

Dviejų etapų STP uždavinys

Planavimo uždaviniuose dažniausiai neįmanoma tiksliai įvertinti atskirų sprendinių realizavimo faktorių. Šiais atvejais gautus sprendinius reikia koreguoti jų realizavimo metu. Dviejų etapų stochastinio programavimo uždavinys leidžia įvertinti neapibrėžtumą ir gauti sprendinius, kurių korekcija reikalautų minimalių išlaidų [1–4].

Formuluojant dviejų etapų STP uždavinį, pirmo etapo kintamųjų vektorius yra pažymėtas *x*. Jis yra randamas anksčiau, nei sužinoma konkreti atsitiktinio parametro reikšmė. Vėliau, atsižvelgiant į atsitiktinio vektoriaus ξ realizaciją, antro etapo sprendinys *y* yra koreguojamas [1].

Dviejų etapų STP uždavinys gali būti formuluojamas taip:

$$F(x) = \min_{x \in D_1 \subset R_+^{n1}} \left[c^T \cdot x + E(f(x,\xi)) \right], \tag{1}$$

kai leistinųjų reikšmių sritis:

$$D_1 = \left\{ x \middle| A \cdot x = b,\, x \in R_+^{n1} \right\}, \tag{2}$$

kur:

$$Q(x,\xi) = \min_y \left\{ q^T \cdot y \middle| W \cdot y + T \cdot x = h,\, y \in R_+^{n2} \right\}, \tag{3}$$

n1 – pirmo etapo kintamųjų skaičius,
m1 – pirmo etapo ribojimų skaičius,
n2 – antro etapo kintamųjų skaičius,
m2 – antro etapo ribojimų skaičius,

c[n1] – tikslo funkcijos koeficientų prie pirmo etapo kintamųjų vektorius,
b[m1] – pirmo etapo ribojimų dešiniųjų pusių vektorius,
q[n2] – tikslo funkcijos koeficientų prie antro etapo kintamųjų vektorius,
h[m2] – antro etapo ribojimų dešiniųjų pusių vektorius,
A[m1×n1] – pirmo etapo ribojimų koeficientų matrica,
T[m2×n1] – antro etapo ribojimų koeficientų prie pirmo etapo kintamųjų matrica,
W[m2×n2] – antro etapo ribojimų koeficientų prie antro etapo kintamųjų matrica.

Tarkime, kad leistinumo sritis D_1 yra netuščia ir aprėžta. Vektorius *h* yra atsitiktinis. Taip pat tarkime, kad antro etapo uždavinio (3) sprendinys ir funkcijos *Q* reikšmės beveik tikrai egzistuoja ir yra aprėžtos.

Kai atsitiktinis parametras ξ yra apibrėžtas diskrečiuoju skirstiniu, kur p_j, $j = 1,\dots N$ žymi kiekvieno scenarijaus tikimybes, *N* yra scenarijų skaičius, galima užrašyti deterministinį formuluotės ekvivalentą:

$$\min_x c^T \cdot x + \sum_{j=1}^{N} p_j \cdot q^T \cdot y^j, \tag{4}$$

kai

$$\begin{aligned} &A \cdot x = b, \\ &T \cdot x + W \cdot y^j = h^j, \\ &x \ge 0,\ \ y^j \ge 0,\ \ j = 1,\dots,N, \end{aligned} \tag{5}$$

kuris susieina į tiesinio programavimo blokinį uždavinį:

$$\begin{cases} A\cdot x & & & & = b, \\ T\cdot x & +W\cdot y^1 & & & = h^1, \\ T\cdot x & & +W\cdot y^2 & & = h^2, \\ \dots & \dots & \dots & \dots & \dots \\ T\cdot x & & & +W\cdot y^N & = h^N, \\ x\ge 0,\ y^1\ge 0,\ y^2\ge 0,\ \dots,\ y^N\ge 0. \end{cases} \tag{6}$$

Sprendžiant anksčiau aprašytą dviejų etapų STP uždavinį, buvo naudojamas modifikuotas L pavidalo (L-shaped) algoritmas. Scenarijai buvo generuojami pagal normalųjį dėsnį $N(\mu_x, \sigma_x)$, kur μ_x yra aritmetinis vidurkis, o σ_x – vidutinis kvadratinis nuokrypis (standartinis nuokrypis).

Modifikuotas dviejų etapų tiesinio stochastinio uždavinio L pavidalo (L-shaped) dekompozicijos algoritmas

Algoritmas yra iteracinis ir susideda iš kelių etapų. Iteracijos kartojamos tol, kol pasiekiamas reikiamas tikslumas. Algoritmo pagrindą sudaro J. R. Bierge aprašytas dekompozicijos metodas [1].

Prieš pirmąją iteraciją priskiriamos pradinės reikšmės: $r = s = v = 0$, kur r – leistinumo ribojimų skaičius, s – optimalumo ribojimų skaičius, v – iteracijos numeris. Toliau aprašyti visi algoritmo žingsniai.

Algoritmo pradžia

1 žingsnis. Sprendžiamas pagrindinis tiesinis uždavinys (*master*):

$$\min_{x,\theta} c^T x + \theta \quad s.t.\ Ax = b, \tag{7}$$

$$D_l x \ge d_l, \quad l = 1,\dots,r, \tag{8}$$

$$E_l x + \theta \ge e_l, \quad l = 1,\dots,s, \tag{9}$$

$x \ge 0,\ \theta \in R$

Jei baigtinis sprendinys x^* neegzistuoja, tai sprendžiamas modifikuotas uždavinys:

$$\min \mathrm{c}^{\mathrm{T}} x + q^T y\cdot u \,\big|\, Ax = b,\ Tx + Wy = h,\ x\ge 0,\ y_0 \ge 0, \tag{10}$$

kur koeficientas u yra labai mažas.

2 žingsnis. Visiems scenarijams k = 1,..., K sprendžiamas tiesinis uždavinys:

$$\begin{aligned} &\min w' = e^T \upsilon^+ + e^T \upsilon^- \\ &s.t.\ Wy + I\upsilon^+ - I\upsilon^- = h_k - T_k x^v, \\ &\quad y,\ \upsilon^+,\ \upsilon^- \ge 0. \end{aligned} \tag{11}$$

Jei kurioms nors k reikšmėms $w' > 0$, tai reikia apibrėžti kintamuosius leistinumo pjūviui:

$$\begin{aligned} D_{r+1} &= (\sigma^v)^T T_k\ , \\ d_{r+1} &= (\sigma^v)^T h_k\ , \end{aligned} \tag{12}$$

pridėti leistinumo ribojimą (8) ir pereiti prie pirmojo žingsnio, kitaip – pereiti prie trečiojo žingsnio.

3 žingsnis. Sprendžiami antro etapo uždaviniai visiems scenarijams (*subproblem*):

$$\min \omega = q_k^T y \,\big|\, Wy = h_k - T_k x^v,\ y\ge 0,\ k = 1,..K. \tag{13}$$

Apibrėžti kintamuosius optimalumo pjūviui:

$$\begin{aligned} E_{r+1} &= \sum_{k=1}^{K} p_k (\pi_k^v)^T T_k\ , \\ d_{r+1} &= \sum_{k=1}^{K} p_k (\pi_k^v)^T h_k\ , \end{aligned} \tag{14}$$

apskaičiuoti $\omega^v = e_{r+1} - E_{r+1} x$.

Jei $\theta^v \ge \omega^v$, reikia sustoti, x^v yra optimalus sprendinys.

Kitaip – pridėti optimalumo ribojimą (9) ir pereiti prie pirmojo žingsnio.

Algoritmo pabaiga

Pirmojoje iteracijoje ribojimų (8), (9) nėra ir θ skaičiavimuose nedalyvauja. Tokiu būdu sprendžiamas uždavinys atrodo taip: min $c^T x | Ax = b, x \geq 0$. Sprendžiant šį uždavinį gaunamas pradinis sprendinys x^*. Jei baigtinis sprendinys x^* neegzistuoja, tai sprendžiamas modifikuotas uždavinys (10), kur antro etapo kintamieji įtraukiami į tikslo funkciją, padauginus naudingumo koeficientus iš pasirinkto mažo dydžio u.

Antrojo žingsnio uždavinys sprendžiamas visiems scenarijams $k = 1,..., K$ tol, kol visos tikslo funkcijos reikšmės nebus lygios nuliui.

Scenarijų agregavimas

Scenarijų agregavimo metodas buvo pasiūlytas Rockafellerio ir Wetso [5]. Vėliau Wetsas aprašė pagrindinius scenarijų analizės ir stochastinio optimizavimo agregavimo principus [6]. Toliau šis metodas buvo taikomas sprendžiant įvairias konkrečias problemas. Taip [7] agregavimo technika yra taikoma sprendžiant dviejų etapų gamybos uždavinį, kai reikia paskirstyti duotą biudžetą produkto komponentėms pirkti. Daugelio etapų optimizavimo problemų neapibrėžtumo sąlygomis scenarijų analizė buvo nagrinėjama [8]. Vėliau scenarijų agregavimo metodas buvo nagrinėjamas [9] – buvo suformuluoti pagrindiniai šio metodo taikymo kriterijai, paremti pavyzdžiais. Žemiau aprašoma scenarijų agregavimo metodo esmė.

Tarkime, kad turime stochastinio optimizavimo uždavinį, apibrėžtą tikimybinėje erdvėje (Ξ, P), kur Ξ yra galimų realizacijų aibė, P yra susijęs tikimybinis skirstinys. Tuomet galima užrašyti:

$$\begin{aligned} &\min E\{f(x,\xi)\} = \int f(x,\xi)dP(\xi) \\ &kai \quad x \in C \subset R^n, \end{aligned} \tag{15}$$

kur f yra tikslo funkcija ir C yra leistinųjų sprendinių aibė, apibrėžta ribojimais.

Kadangi modelis priklauso nuo atsitiktinio vektoriaus ξ ir jo tikimybinio skirstinio P, jis negali būti naudojamas kaip tinkama modeliavimo priemonė, kai prieinama tik ribota informacija apie atsitiktinių parametrų pasiskirstymą. Tokiais atvejais dažniausiai naudojama scenarijų analizė. Tarkime, kad neapibrėžtumas yra modeliuojamas keleto scenarijų:

$$S = \{s^1, \ldots, s^L\},$$

ir kiekvienam scenarijui $s \in S$ randamas žemiau pateiktos subproblemos (P_S) sprendinys:

$$\begin{aligned} &\min f(x,s) \\ &visiems \quad x \in C_s \subset R^n. \end{aligned} \tag{16}$$

Tarkime, kad mes žinome, kaip gauti kiekvieno atskirai paimto scenarijaus sprendinį. Problema yra ta, kad neaišku, kaip dirbti su skirtingais S priklausomų sprendinių vektoriais, sprendžiant jų sujungimo ir bendro sprendinio gavimo problemą [8].

Tarkime, kad optimalus sprendinys egzistuoja kiekvienam $s \in S$. Tuomet optimalų scenarijaus sprendinį galima pažymėti taip:

$$x^s \in \arg\min\{f(x,s) \mid x \in C_s\}.$$

Kai scenarijų sprendiniai visiems s iš S yra apskaičiuoti, jie analizuojami siekiant rasti bendras tendencijas ar sprendinių klasterius, nustatyti, kaip sprendinys būtų apskaičiuojamas, jei scenarijus s' iš tikrųjų įvyktų. Vidutinis sprendinys apskaičiuojamas sprendinius x^s padauginus iš scenarijų tikimybių; gauti vidutiniai sprendiniai analizuojami toliau. Galutinis analizės tikslas yra gauti vieną sprendinį, kuris gali būti naudojamas priimant sprendimą.

Tegu p_s, $s \in S$ yra scenarijų tikimybės. Konstruojamas įvertis, pažymintis vidutinį sprendinį:

$$\hat{x} := \sum_{s \in S} p_s x^s, \tag{17}$$

kur $\{x^s\} = \{x^1, \ldots, x^L\}$ yra scenarijų problemų sprendiniai. Čia turima omenyje, kad sprendinys $\hat{x}$ nepriklauso nuo scenarijų, t. y. mes galime žiūrėti į sprendinį, kuris, apskritai imant, labiau reaguos į visus dažniausius galimus įvykius nei į ypatingus atskirai paimtų scenarijų sprendinius x^s. Tačiau $\hat{x}$ nebūtinai yra įmanomas. Sprendinys x yra priimtinas, jeigu jis yra įmanomas kiekvieno konkretaus scenarijaus atveju, t. y. $x \in C_s$ visiems s iš S arba ekvivalentiškai $x \in \cap_{s \in S} C_s$.

Tuomet galime apibrėžti stochastinio optimizavimo problemą, kai kiekvienas scenarijus s yra susijęs su scenarijaus tikimybe p_S:

$$\begin{aligned} &\min \sum p_s f(x, s^s) \\ &kai \quad x \in \cap_s C_s \end{aligned} \tag{18}$$

Optimalus problemos sprendinys yra x^*.

Skaičiavimų rezultatai

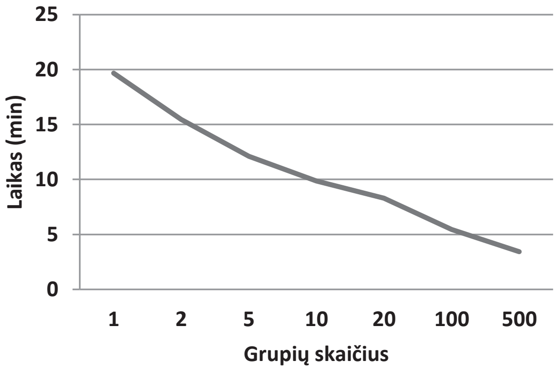
Skaičiavimai buvo atliekami kompiuteriu, kurio parametrai yra: Intel(R) Core(TM) i7-4500U CPU @ 1,80 GHz 2.4 GHz, 8.00GB, x64-based processor. Programa realizuojama *Microsoft Visual Studio 2010* C++ kalba, naudojant IBM ILOG CPLEX optimizavimo paketą. Testinis dviejų etapų uždavinys turi 20 kintamųjų ir 10 ribojimų pirmame etape ir 30 kintamųjų bei 20 ribojimų antrame. Scenarijai buvo padalinti į 1, 2, 5, 10, 20, 100 ir 500 grupių. Skaičiavimai pakartoti 100 kartų kiekvieno padalinimo atveju. Kiekvieno atvejo suvidurkinti rezultatai pateikti 1 lentelėje.

1 lentelė. ***Skaičiavimų rezultatai***

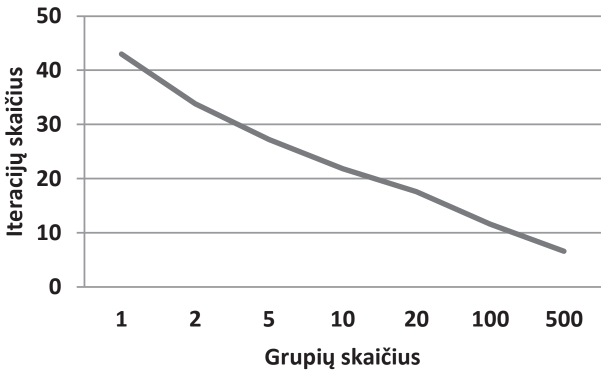
Grupių sk.	1	2	5	10	20	100	500
Laikas (min.)	19,66	15,45	12,1	9,86	8,29	5,45	3,43
Iteracijų sk.	43,1	33,8	27,2	21,8	17,6	11,6	6,6

Vidutinė pagrindinio (*master*) uždavinio tikslo funkcijos reikšmė yra TF = 182,58.

Iš rezultatų matome, kad didinant grupių skaičių skaičiavimo laikas ir iteracijų skaičius, reikalingas optimaliai to paties tikslumo reikšmei pasiekti, mažėja. Laiko ir iteracijų skaičiaus priklausomybė nuo grupių skaičiaus pavaizduota 1 paveiksle, atitinkamai (A) ir (B).



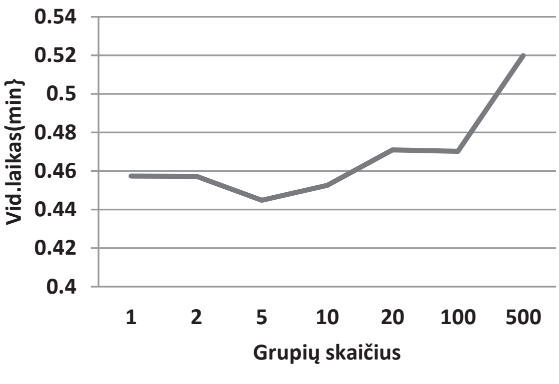
(A)



(B)

1 pav. Laiko (A) ir iteracijų skaičiaus (B) priklausomybė nuo grupių skaičiaus

2 paveiksle pavaizduota vidutinio vienos iteracijos skaičiavimo laiko priklausomybė nuo grupių skaičiaus.



2 pav. Vidutinio vienos iteracijos skaičiavimo laiko priklausomybė nuo grupių skaičiaus

Iš grafiko matyti, kad, didinant grupių skaičių, vienos iteracijos skaičiavimo laikas didėja. Taip yra dėl to, kad, didinant grupių skaičių, atitinkamai padidėja pagrindinio (*master*) uždavinio ribojimų skaičius.

Išvados

1. Darbe sukurtas tolydžiojo mato diskretizavimu besiremiantis dekompozicijos metodas STP uždaviniams spręsti, kai stochastiniai kintamieji aprašyti absoliučiai tolydžiuoju tikimybiniu dėsniu.

2. Gauti rezultatai parodė, kad, didinant scenarijų grupių skaičių, iteracijų skaičius sumažėja (žr. 1 pav. B), drauge trumpėja bendras skaičiavimo laikas tam pačiam tikslumui pasiekti (žr. 1 pav. A).
3. Didinant grupių skaičių, vienos iteracijos skaičiavimo laikas ilgėja. Taip yra dėl to, kad, didinant grupių skaičių, atitinkamai padidėja pagrindinio (*master*) uždavinio ribojimų skaičius.

Literatūra

1. Birge J. R., Louveauxl F. V., 2011, Introduction to Stochastic Programming. *Springer*. New York.
2. King A. J., Wallace S. W., 2012, Modeling with Stochastic Programming. *Springer Series in Operations Research and Financial Engineering*. *Springer*. New York.
3. Wets R. J.-B., 1988, Large scale linear programming techniques. *Numerical Techniques in Stochastic Optimization*. P. 65–94.
4. Wets R. J.-B., 1989, Stochastic programming. *Handbooks in Operations Research and Management Science*. Vol. 1: *Optimization*. P. 573–629.
5. Rockafellar R. T., Wets R. J.-B., 1987, The principle of scenario aggregation in optimization under uncertainty, *Working paper WP-87-119*, Math. Oper. Res.
6. Wets R. J.-B., 1989, The aggregation principle in scenario analysis and stochastic optimization. *Algorithms and Model Formulations in Mathematical Programming*. P. 91–113.
7. Jönsson H., Jörnsten K., Silver E. A., 1993, Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint. *European Journal of Operational Research*. Vol. 68, issue 2. P. 196–211.
8. Rockafellar R. T., Wets R. J.-B., 1991, Scenarios and policy aggregation in optimization under uncertainty, *Math. Oper. Res*. 16. P. 119–147.
9. Cambou M., Filipović D., 2015, Model Uncertainty and Scenario aggregation. Swiss Finance Institute Research Paper No. 14–38. Prieiga per internetą: <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2441328>.

Summary

INVESTIGATION OF AGGREGATION PROBLEM IN TWO-STAGE STOCHASTIC LINEAR PROGRAMMING

A. Ušpurienė, L. Sakalauskas

The paper focuses on a two-stage stochastic linear programming problem when second stage variables depend on a random parameter with a normal (Gaussaian) distribution. Modified L-shaped algorithm using the optimization software CPLEX was applied to solve this problem. The aggregation scenario to reduce the number of iterations of the optimization process, the amount of resources to be used and the time required to produce an optimal solution have been analyzed in the paper. The efficiency of aggregation method was tested using statistical simulation method.

Keywords: stochastic programming, scenario generation, aggregation, optimization, L-shaped method.

Santrauka

DVIEJŲ ETAPŲ STOCHASTINIO TIESINIO PROGRAMAVIMO UŽDAVINIO AGREGAVIMO TYRIMAS

A. Ušpurienė, L. Sakalauskas

Darbe nagrinėjamas dviejų etapų stochastinio tiesinio programavimo uždavinys, kai antro etapo kintamieji priklauso nuo atsitiktinio parametro, pasiskirsčiusio pagal normalųjį (Gauso) dėsnį. Sprendžiant šį uždavinį, pritaikytas modifikuotas L pavidalo (L-shaped) algoritmas, realizuotas naudojant CPLEX optimizavimo paketą. Darbe nagrinėjamas scenarijų agregavimo metodas, siekiant sumažinti optimizavimo proceso iteracijų skaičių, naudojamų resursų kiekį ir optimalaus sprendinio gavimo skaičiavimo laiką. Agregavimo metodo efektyvumas ištirtas statistinio modeliavimo būdu.

Prasminiai žodžiai: stochastinis programavimas, scenarijų generavimas, agregavimas, optimizavimas, L pavidalo (L-shaped) metodas.

Įteikta 2016-02-25
Priimta 2016-05-10