

10 Lao majandamise „kõik või mitte midagi“ mudel

Vaatleme mudelit, kus näiteks masinaliin toodab pidevalt ja ühtlaselt mingit toodangut, mida kogutakse lattu. Laost võetakse kaupa, kui tuleb vastav tellimus (ostu soov) kliendi poolt. Juhumata tootmisel võib lattu korraka koguneda liiga palju või hoopis vähe toodangut. Toodangu ladustamisega kaasnevad seejuures kulud, näiteks lao tühendamisel (tööjõu-, transpordikulu), laos hoidmisel (üüri-, elektri-, valve-, tööjõukulud).



„Kõik või mitte midagi“ mudel on selline, kus klient ostab toodangu ainult siis, kui talle suudetakse pakkuda kliendi valitud kogus. Kui laos on vähem toodangut kui tellimus, siis selle mudeli järgi klient tellimust ei esita ja otsib endale teise pakkuja. Sellega seoses võib esineda veel nn. alatoodangust tulenev kulu (nn. teenimisvõimaluse mittekasutamine), mis võib oma olemuselt olla ka reaalne (näiteks leppetrahv või hüvitis). Teisalt, liiga suur toodang toob kaasa uued lao kulud ja need võivad olla küllalt suured. Küsimus on, kuidas kulutusi optimeerida.

10.1 Modelleerimine Volterra II liiki integraalvõrrandiga

Vaatleme lao täitmise ja tühendamise „kõik või mitte midagi“ tüüpi pidevat mudelit (vt. [1]), milles teatud tihedusfunktsioon leitakse võrrandist

$$v(x) - v(b) = \lambda \cdot \int_x^b (e^{-\mu \cdot (s-x)} - e^{-\mu \cdot s}) \cdot v(s) ds, \quad x \in [0, b] \quad (10.1)$$

- $v = v(x)$ on lao täitumise tihedusfunktsioon, mis näitab lao täitumistaseme x keskmist ületamiste arvu.
- argumenti x võib olenevalt kontekstist vaadelda kui tellimuse suurust või siis lao täitumist. Ladu tühendatakse täielikult (matemaatiliselt) taseme $x = b$ juures.

- Mudel on tuletatud Poissoni jaotusest intensiivsusega λ , milles juhuslikud tellimused on keskvaartusega $\frac{1}{\mu}$.
- Tuuma liige $e^{-\mu \cdot (s-x)} - e^{-\mu \cdot s}$ näitab tõenäosust $P(s-x \leq T < s)$ tellimuse T rahuldamiseks.
- Lao täitumise/tühjenemise mudeli tihedusfunktsioon v peab rahuldama tingimust

$$\int_0^b v(s) ds = 1. \quad (10.2)$$

Lao majandamise kulud

Toodangu laos hoidmisele, alatootmisest tingitud ja lao tühjenemisele minevad kulud kokku (rahaühikut ajaühiku kohta) on leitavad seosega

$$C(b) = H \cdot S(b) + p \cdot \lambda \cdot \left(\frac{1}{\mu} + S(b) \right) \cdot \int_0^b (1 - G(s)) \cdot v(s) ds + k \cdot v(b), \quad (10.3)$$

kus $S(b)$ ja $G(s)$ on antud valemitega

$$S(b) = \int_0^b s \cdot v(s) ds, \quad G(s) = 1 - e^{-\mu \cdot s}.$$

- H on toote laos hoidmise kulud (ühiku kohta ühes ajaühikus).
- p on alatootmisest tingitud kulud.
- k on lao tühjenemisele minevad kulud.
- $\lambda \cdot \int_0^b (1 - G(s)) \cdot v(s) ds$ on tellimuste arv, mis jääb lao alataitumise tõttu rahuldamata.

Mudelit saab kasutada ka näiteks vaksineerimise, kinnisvaraturu jne. modelleerimiseks. Mudel ei arvesta „võlgu“ tootmist.

Viited

- [1] O. Berman, M. Parlar, D. Perry, M. J. M. Posner. Production/Clearing Models Under Continuous and Sporadic Reviews. Meth. Comp. Appl. Prob., 7, 203-224, 2005.

10.2 II liiki lineaarse Volterra integraalvõrrandi lahendamine kvadratuursummade meetodiga

1. Otsime võrrandi

$$v(x) = \int_x^b K(x, s) v(s) ds + f(x), \quad x \in [0, b],$$

lahendit $v = v(x)$.

2. Loome ühtlase vahemikega sõlmed

$$0 = x_1 < x_2 < \dots < x_N = b, \quad x_i = h \cdot (i - 1), \quad i = 1, \dots, N, \quad h = \frac{b}{N - 1}.$$

3. Esimene võrrand tuleb tingimusest (10.2). Kasutame trapetsvalemit samades sõlmedes,

$$\frac{h}{2} \cdot v_1 + h \cdot v_2 + \dots + h \cdot v_{N-1} + \frac{h}{2} \cdot v_N = 1.$$

4. Kirjutame võrrandi (10.1) välja $N - 1$ esimeses sõlmes,

$$v_i - \int_{x_i}^b K(x_i, s) v(s) ds = f(x_i), \quad i = 1, \dots, N - 1.$$

Asendades integraalid trapetsvalemitega, saame lineaarvõrrandite süsteemi $Av = F$ ülejäänud võrrandid:

$$v_i - \sum_{j=i}^N w_j \cdot K(x_i, x_j) \cdot v_j - v_N = 0, \quad i = 1, \dots, N - 1,$$

kus

$$w_j = \frac{h}{2}, \quad w_{j+1} = h, \quad \dots, \quad w_{N-1} = h, \quad w_N = \frac{h}{2}.$$

Esitame siinjuures süsteemi $Av = F$ skemaatiliselt:

$$\begin{pmatrix} 1 - a_{11} & -a_{12} & -a_{13} & \dots & -a_{1,N-1} & -(a_{1,N} + 1) \\ 0 & 1 - a_{22} & -a_{23} & \dots & -a_{2,N-1} & -(a_{2,N} + 1) \\ 0 & 0 & 1 - a_{33} & \dots & -a_{3,N-1} & -(a_{3,N} + 1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 - a_{N-1,N-1} & -(a_{N-1,N} + 1) \\ h/2 & h & h & \dots & h & h/2 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{N-1} \\ v_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

kus

$$a_{ij} = w_j \cdot K(x_i, x_j), \quad i = 1, \dots, N - 1, \quad j = i, \dots, N.$$

Tavaliselt täidetakse maatriks A kahe for-tsükli abil, mida võib ka siinkohal teha. Arvutusaaja mõttes annab olulist võitu ühe for-tsükli kasutamine ja vektoriseerimine:

```

AA( i , i:N ) = w( i : N ) .* K( x( i ) , x( i:N ) )
// NB! Sellisel juhul peab tuum K olema hoolikalt defineeritud
//...
AA = eye( AA ) - AA
// Viimase veeru ja rea saame täiendada eraldi
AA = sparse(AA) // umfpack käsu jaoks
yy = umfpack(AA, '\', FF) // Töötab kiiremini kui linsolve
// (hõreda matriksi AA jaoks)

```

10.3 Praktikum ülesanne

1. (3p) Lahendada võrrand (10.1) trapetsvalemiga abil $b = 3$ korral.
2. Kasutame parameetrite väärtusi

$$\lambda = 5, \quad \mu = 10, \quad H = 1, \quad p = 2, \quad k = 4.$$

3. Taha tihedusfunktsioonist v graafik.
4. (1p) Leida vastav kulu $C(b)$, $b = 3$.
5. (1p) Leida optimaalne toodangu tase b_* ja sellele vastav minimaalne kulu $C(b_*)$. Selleks võib näiteks lahendada võrrandit (10.1) 10-20 korda ühes for-tsükli, muutes b väärtusi, näiteks $b = 0.2, 0.4, 0.6, \dots$. Samas tsükli saab leida ka b väärtusele vastava kulu $C(b)$.
6. Lao kulude $C(b)$ arvutamisel, asendame integraalid näiteks trapetsvalemiga, kasutades samu sõlmi x_1, \dots, x_N . Näiteks,

$$w = \left[\frac{h}{2}, h, \dots, h, \frac{h}{2} \right],$$

sõlmevektori x ja lahendi vektori v korral

$$\int_0^b s \cdot v(s) ds \approx \sum_{j=1}^N w_j \cdot \left(x_j \cdot v(x_j) \right).$$

Viimast saab leida lihtsalt summa või „inttrap“ käsuga:

```

sum(w.*x.*v') // summa üle vektori
inttrap(x, x.*v') // SciLab'i enda sisevahend

```