OPTIMIZAVIMAS IR TAIKYMAI

Jonas Mockus

Konsultacijos:
jonas@optimum.mii.lt

Mokymo priemonės tai tinklapiai:
$http://soften.ktu.lt/\tilde{}mockus$
$http://optimum.mii.lt/\tilde{}jonas$
$http://mockus.org/optimumlt$
$http://mockus.org/optimum$
Pastarasis skirtas tarptautiniam naudojimui.

Vadovėlis:
A. Žilinskas
"Matematinis Programavimas"
Kaunas, VDU, 1999

OPTIMALUMAS

Tikslo funkcija

$$f(x),\ x = (x_1, ..., x_m). \tag{1}$$

Globalus minimumas

$$f(x_A) \leq f(x)\ x \in A, \tag{2}$$

arba

$$x_A = \arg\min_A f(x), \tag{3}$$

kur A leistina sritis.

Lokalus minimumas

$$f(x_\epsilon) \leq f(x)\ x \in \epsilon, \tag{4}$$

kur ϵ taško x_ϵ aplinka.

Diskretus optimizavimas

Kai x_i diskretus.

Tiesinis programavimas

Kai

$$f(x) = \sum_{i=1}^{m} c_i x_i, \quad (5)$$

$$A: \sum_{i=1}^{m} a_{ij} x_i \geq b_j, \ j = 1, ..., m, \ x_i \geq 0. \quad (6)$$

Iškilas programavimas

Kai tikslo funkcija $f(x)$ ir leistina aibė A yra abi iškilos.

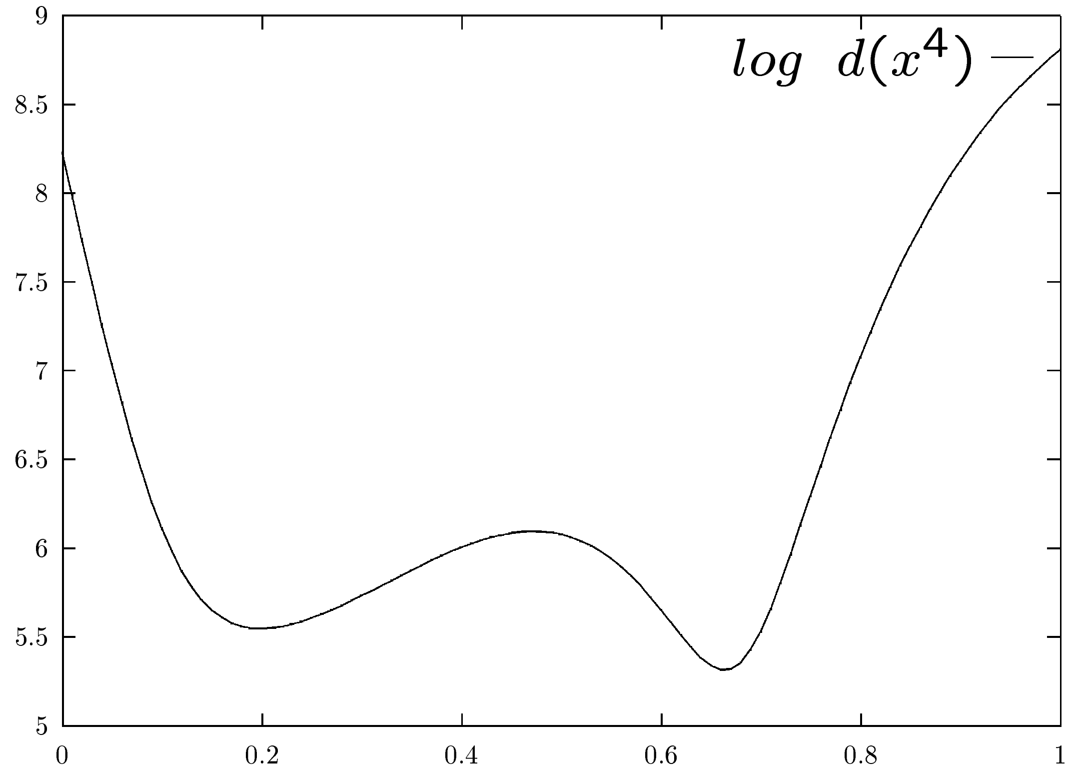


Figura 1: Globalus ir lokalus minimumai

Lokalus minimumas $x = 0.2$,
globalus minimumas $x = 0.66$.
Funkcija daugia-ekstremali intervale [0.0,1.0],
tačiau, iškila intervaluose [0.0, 0.3] ir [0.6,0.8],
bei uni-modali intervaluose [0.0,0.48] ir [0.48,1.0].

TAIKYMAI LOŠIMŲ TEORIJOJE IR RINKOS EKONOMIKOJE

Pusiausvyros ieškojimas

Pusiausvyra tai sutartis kurios laužyti neverta.
Pavyzdžiai:
konkurencijos, inspekcijos ir dvikovos modeliai.

Prognozė

Pavyzdžiai:
valiutų ir akcijų kursai,
"call-center" užsakymai.

Optimalus investavimas

Pavyzdžiai:
Portfelio uždavinys bei optimalus draudimas.

DISKRETAUS IR DINAMINIO PROGRAMAVIMO TAIKYMAI

Tvarkaraščių optimizavimas
Pavyzdžiai:
siuvykla ir mokykla.

Nuoseklūs sprendimai
Pavyzdžiai:
jaunikio, kompiuterio,
bei automobilio issirinkimas.

NASH'O PUSIAUSVYRA

Severių konkurencija, Nash'o variantas

Serverio i pelnas

$$u_i = a_i y_i - x_i, \ i = 1, ..., m, \tag{7}$$

$$\sum_{i=0}^{m} a_i = a, \tag{8}$$

kur a užsakymų intensyvumas.

Užsakymas renkasi serverį i^*, jei

$$h_{i^*} \leq h_i, \ i = 0, ..., m, \tag{9}$$

$$h_i = y_i + \gamma_i, \ \gamma_i = n_i / x_i. \tag{10}$$

kur

y_i aptarnavimo kaina,

γ_i laukimo kaina.

Optimali sutartis

Sutartis optimali, jei jos neapsimoka laužyti. Tegu sutartinis vektorius $(\bar{y}_i, \bar{x}_i)$, tada "apgavinis" vektorius

$$(\bar{\bar{y}}_i, \bar{\bar{x}}_i) = \arg\max_{(y_i, x_i)} u_i(y_i, x_i, \bar{y}_j, \bar{x}_j,\ j \neq i), \quad i = 1, ..., m. \tag{11}$$

Sutartis $x = (\bar{y}_i, \bar{x}_i,\ i = 1, ..., m)$ optimali, kai

$$\min_x f(x) = 0, \tag{12}$$

kur

$$f(x) = \sum_{i=1}^{m} ((\bar{\bar{y}}_i - \bar{y}_i)^2 + (\bar{\bar{x}}_i - \bar{x}_i)^2). \tag{13}$$

$f(x)$ minimizuojama globaliu optimizavinmo metodu.

Stabilios koalicijos

Koalicija stabili, jei neapsimoka iš jos pasitraukti.
Tegu $m = 3$.
Žymėkim
S_1 "koaliciją" iš vieno serverio,
S_2 koaliciją iš dviejų simetriškų serverių,
S_3 monopolinę koaliciją.
Dalinant visų koalicijos narių
maksimalų garantuotą pelną po lygiai
$u_i(S_3) = (u_i + u_j + u_k)/3$,
$u_i(S_2) = (u_i + u_j)/2$,
$u_i(S_1) = u_i$.
Monopolis bus stabilus kai
$u_i(S_3) > \max\{u_i(S_2), u_i(S_1)\}$.
Laisva individuali konkurencija bus stabili kai
$u_i(S_1) > \max\{u_i(S_2), u_i(S_3)\}$.
Pora bus stabili kai
$u_i(S_2) > \max\{u_i(S_1), u_i(S_3)\}$.

Serverių konkurencija, Walras'o variantas

Serverio i pelnas

$$u_i = a_i y_i,\ \ i = 1, 2. \tag{14}$$

(15)

Čia laukimo kaina

$$\gamma_i = n_i / w_i, \tag{16}$$
$$w_i = c_{0i}(1 - e^{-c_{ii} x_{ii} - c_{ij} x_{ij}}), \tag{17}$$

kur w_i serverio i galingumas,
x_{ii} savi resursai, x_{ij} partnerio j resursai,
c resursų efektyvumas.
Resursų balanso sąlyga:

$$x_{ii} + x_{ij} = b_i,\ \ i = 1, 2, \tag{18}$$

(19)

kur b_i serverio i resursas.

Inspektoriaus uždavinys

Inspektoriaus pelnas

$$u(i,j) = \begin{cases} p_i g_i q_j, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \tag{20}$$

Brakonieriaus pelnas

$$v(i,j) = \begin{cases} -p_i g_j q_j + (1-p_i) g_j q_j, & \text{if } i = j \\ g_j q_j, & \text{if } i \neq j, \end{cases} \tag{21}$$

kur p_i sugavimo tikimybė miške i,
q_i nušovimo tikimybė,
g_i "briedžio" kaina.
Vidutiniai pelnai

$$U(x,y) = \sum_{i,j} x_i u(i,j) y_j, \tag{22}$$

$$V(x,y) = \sum_{i,j} x_i v(i,j) y_j, \tag{23}$$

kur
x_i, y_i inspektoriaus ir brakonieriaus tikimybės eiti į "mišką" i.

Optimali inspekcija

Pelnų suvienodinimas

$$\sum_{j=1}^{m} u(i,j)y_j^0 = U,\ \ i = 1, ..., m \tag{24}$$

$$\sum_{i=1}^{m} v(i,j)x_i^0 = V,\ \ i = 1, ..., m$$

$$\sum_{i=1}^{m} x_i = 1,\ \sum_{i=1}^{m} y_i = 1$$

Jei sprendinys x, y tenkina sąlygąs
$x_i \geq 0,\ \ y_i \geq 0,\ \ i = 1, ..., m,$
tai jis nurodo optimalią inspekciją.

Jei ne, tai tikrinam pusiausvyros sąlygas visuose pelno lentelių langeliuose tikslu surasti pusiausvyros situaciją grynų strategijų aibėje.

Inspekcijos pavyzdys

Tegu $q_i = g_i = 1$. Tada iš (20)(21)

$$u(i,j) = \begin{cases} p_j, & \text{jei } i = j \\ 0, & \text{jei ne,} \end{cases} \tag{25}$$

ir

$$v(i,j) = \begin{cases} -p_i + (1 - p_i), & \text{jei } i = j \\ 1, & \text{jei ne.} \end{cases} \tag{26}$$

Iš čia ir (24)

$$p_j y_j = U,\ j = 1, ..., m \tag{27}$$

$$\sum_{i \neq j} x_i + (1 - 2p_j) x_j = V,\ j = 1, ..., m \tag{28}$$

$$\sum_j y_j = 1,\ \sum_i x_i = 1,\ y_j \geq 0,\ x_i \geq 0$$

Kai $m = 2$ iš (27)

$$y_1 = p_2/(p_1 + p_2), \tag{29}$$

$$y_2 = p_1/(p_1 + p_2), \tag{30}$$

iš (28)

$$x_1 = p_2/(p_1 + p_2), \tag{31}$$

$$x_2 = p_1/(p_1 + p_2). \tag{32}$$

Pavyzdžio sprendinys

$$u^* = U = p_1 p_2/(p_1 + p_2),$$
$$v^* = V = p_1 + p_2 - 2p_1 p_2/(p_1 + p_2). \quad (33)$$

Kai $p_1 = 1/3, p_2 = 2/3$

$x1^* = y_1^* = 2/3,\ x_2^* = y_2* = 1/3,\ u^* = 2/9,\ v^* = 5/$

Bendresniu atvėju

$$y_i = x_i = \prod_{k \neq i} p_k / (\sum_i \prod_{k \neq i} p_k), \quad (35)$$

$$(36)$$

kur $\prod_{k \neq i} p_k$ tai sandauga visų p_k iškyrus p_i.

Dvikova

Kaunasi du skraidantys objektai.
Jų judesius aprašo dif-lygtys

$$dz(t)/dt = az(t), \tag{37}$$

$$dw(\tau)/d\tau = bw(\tau), \tau = 2 - t, \tag{38}$$

kurių sprendiniai

$$z(t) = z_0 e^{at},\ \ w(\tau) = w_0 e^{b\tau}. \tag{39}$$

Pataikymo tikimybė

$$p(t) = 1 - d(t)/D, \tag{40}$$

kur $d(t)$ atstumas tarp objektų momentu t,
D maksimalus atstumas.

Optimali dvikovos strategija

Naudingumo funkcijos

$$U(t_1,t_2) = \begin{cases} p(t_1) - (1 - p(t_1)), & \text{if } t_1 < t_2, \\ -p(t_2) + (1 - p(t_2)), & \text{if } t_2 < t_1, \\ 0, & \text{if } t_2 = t_1, \end{cases}$$
$$V(t_1,t_2) = \begin{cases} p(t_2) - (1 - p(t_2)), & \text{if } t_2 < t_1, \\ -p(t_1) + (1 - p(t_1)), & \text{if } t_1 < t_2, \\ 0, & \text{if } t_2 = t_1. \end{cases}$$

Čia optimalūs šovimo momentai kai

$$p(t_1) = p(t_2) = 0.5. \tag{41}$$

Optimalius pradinius auksčius z_0, w_0 bei kilimo greičius a, b randam iš pusiausvyros sąlygų naudodami mišrias strategijas
kurias nustatom tiesiniu programavimu.

Ekonominė dvikova
Dinaminė dviejų severių konkurencija

Serverių pelno funkcijos

$$U_i(T) = \int_{t_0}^{T} u_i(t)dt, \tag{42}$$

$$u_i(t) = a_i(t)y_i(t) - x_i(t), \tag{43}$$

kur
$u_i(t)$ serverio i pelnas momentu t,
$a(t) = \sum_i a_i(t) =$ užsakymų srauto intensyvumai.
Aptarnavimo kaina $y_i(t)$ ir išlaidos $x_i(t)$ aprašomi dif-lygtimis

$$dy_i(t)/dt = ay_i(t), \tag{44}$$

$$dx_i(t)/dt = bw_i(t). \tag{45}$$

Tikslas- rasti pradines reikšmes $y_i(0), x_i(0)$, bei kitimo greičius a, b,
nusakančias maksimalų garantuotą pelną.
Uždvinio esmė ta, kad severis i bankrutuoja momentu t^*, jei $U_i(t^*) < -U^*$.
Tada likęs severis gauna monopolinį pelną.

Portfelio uzdavinys, Optimalus investavimas

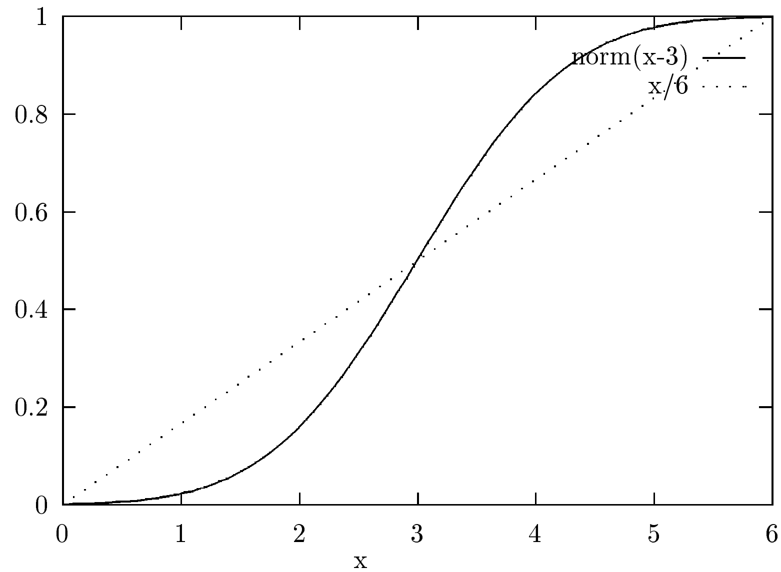


Figura 2: Naudingumo funkcija $u(x)$.

turtingo asmens ...
vidutinio asmens -
investuojamas kapitalas x
visas kapitalas 6.0
rizikos zona [0.0, 3.0]
atsargumo zona [3.0,6.0]

Loterija

Kapitalo $0 \leq C \leq 1$ naudingumą $u(C)$ nustatom pagal loteriją

$$C \sim \{pA + (1-p)B\}, \tag{46}$$

kur

išlošimo C naudingumas $u(C) = p$,

p tikimybė išlošti viska $A = 1$,

$1 - p$ tikimybe nieko neišlošti $B = 0$.

Ženklas $\sim$ nusako "svyravimo tikimybę" p, kada asmeniui neaišku kas geriau:

- ar išsaugoti turimc a kapitalą C,
- ar dalyvauti loterijoje tikintis išlošti A.

Laukiamas naudingumas

$$U(x) = \sum_{k=1}^{M} u(y^k) p(y^k), \tag{47}$$

$$\tag{48}$$

kur
$y^k = \sum_{i=1}^{m} \delta_i c_i x_i$ kapitalas kurį gausim po metų,
$u(y^k)$ kapitalo y^k naudingumas,
x_i kapitalas įdetas dabar į banką i,
$c_i = 1 + \alpha_i$, α_i banko i palūkanos,
$\delta_i = 1$ jei bankas i nebankrutuoja,
$\delta_i = 0$ jei bankas i bankrutuoja
$p(y^k)$ tikimybe, kad gausim kapitalą y^k,
pavyzdžiui,
$p(y^1) = p_1 \prod_{i \neq 1}(1 - p_i)$,
Čia $y^1 = c_1 x_1$,
p_i tikimybė, kad bankas i nesubankrutuos.

Optimalus draudimas

Laukiamas naudingumas

$$U(x) = \sum_{k=1}^{M} u(y^k) p(y^k).$$

Čia $p(y^k)$ tikimybe, kad po metų turėsim turtą $y^k = \sum_{i=1}^{m} c_i(x_i)$, $u(y^k)$ turto y^k naudingumas.

$$c_i(x_i) = \begin{cases} z_i - a_i x_i, & \text{jei } \delta_i = 1 \\ (1 - a_i) x_i, & \text{jei } \delta_i = 0, \end{cases} \qquad (49)$$

kur z_i objekto i vertė ,
$a_i x_i$ objekto i draudimo mokestis,
$x_i \leq z_i$ objekto i draudimas.
$\delta_i = 0$ jei objektas i žus,
$\delta_i = 1$ jei nežus,
$p_i = P\{\delta_i = 1\}$ tikimybė, kad objektas i nežus.
Pavyzdys:
$p(y^1) = p_1 \prod_{i \neq 1} (1 - p_i)$,
$y^1 = c_1(x_1) + \sum_{i=2}^{m} c_i(x_i)$, kur
$c_1(x_1) = z_1 - a_1 x_1$,
$c_i(x_i) = (1 - a_i) x_i, \; i = 2, ..., m.$

Prognozė

ARMA modelis

Tai auto-regresijos slenkančio vidurkio modelis

$$w_t = \sum_{i=1}^{p} a_i w_{t-i} + \sum_{i=1}^{q} b_j \epsilon_{t-j} + \epsilon_t. \tag{50}$$

Pavyzdžiui ,
w_t akcijų kursas rytoj
w_{t-1} akcijų kursas šiandien,
ϵ_t atsitiktinė paklaida rytoj,
a_i, b_j įtakos koeficientai,
juos nustatom minimizuodami

$$f(x) = \sum_{t=1}^{T} \epsilon_t^2, \tag{51}$$

kur $x = (x_1, ..., x_{p+q}),\ x_i = a_i,\ i = 1, ..., p,$
$x_i = b_{i-p},\ i = p + 1, ..., p + q.$

Akcijų biržos lošimas

Vienas akcinė bendrovė, du klientai $i = 1, 2$,
Momentu t klientas i perka vieną akcija, jei

$$z(t) \leq z_i^{min}(t), \tag{52}$$

kur
$z(t)$ akcijų kursas laiku t,
$z_i^{min}(t)$ pirkimo "slenkstis".
Momentu t klientas i parduoda akcija, jei

$$z(t) \geq z_i^{max}(t), \tag{53}$$

kur
$z_i^{max}(t)$ pardavimo slenkstis.

Monte Karlo modelis

Akcijų kursas momentu $t+1$ nusakomas pagal aukščiausisią pirkimo kainą momentu t bei žemiausią pardavimo kainą momentu t:

$$z(t+1) = \; =$$

$$\begin{cases} z(t)+\epsilon(t+1), & \text{if } z^{min}(t) \leq z(t) \leq z^{max}(t), \\ z^{min}(t)+\epsilon(t+1), & \text{if } z(t) \leq z^{min}(t), \\ z^{max}(t)+\epsilon(t+1), & \text{if } z(t) \geq z^{max}(t). \end{cases}$$

Čia

$$z^{min}(t) = \max_i \; z_i^{min}(t), \tag{54}$$

$$z^{max}(t) = \min_i \; z_i^{max}(t). \tag{55}$$

Laukiamas akcijos pirkimo pelnas

Laukiamas santykinis akcijos pirkimo pelnas:

$$\Delta_i(t+1) = (\beta_i(t+1) + \delta(t+1) - \alpha(t+1))\ z(t). \quad (56)$$

Čia

$$\beta_i(t+1) = \frac{z_i(t+1) - z(t)}{z(t)}, \quad (57)$$

kur $z_i(t+1)$ akcijų kursas
kurio tikisi klientas i laiku $t+1$

$$\delta(t+1) = \frac{d(t+1)}{z(t)}, \quad (58)$$

kur $d(t+1)$ akcijos dividendai
kurių tikisi klientas i laiku $t+1$

$$\alpha(t+1) = a(t+1)/z(t), \quad (59)$$

kur $a(t+1)$ terminuoto indėlio palūkanos laiku $t+1$.

Pirkimo-pardavimo slenkstis

Pirkimo slenkstis

$$z_i^{min}(t) = k_{buy}\ \Delta_i(t),\ \ k_{buy} > 1. \qquad (60)$$

Pardavimo slenkstis

$$z_i^{max}(t) = k_{sell}\ \Delta_i(t),\ \ k_{sell} < 1. \qquad (61)$$

Akcijų skaičius pas klientą i laiku t

$$N_i(t+1) = \begin{cases} N_i(t), & \text{if } z_i^{min}(t) \le z(t) \le z_i^{max}(t), \\ N_i(t)+1, & \text{if } z(t) \le z_i^{min}(t), \\ N_i(t)-1, & \text{if } z(t) \ge z_i^{max}(t). \end{cases}$$

Klientų prognozės

Kliento i pelnas laike $T_0 \leq t \leq T$

$$u_i(T, T_0) = N_i(T) z_T - N(T_0) z_{T_0} - \sum_{t=T_0}^{T} (N_i(t+1) - N_i(t)) \ z(t).$$

Pelnas u_i priklauso nuo
prognozės $\beta_i(t+1)$ tikslumo ir
nuo atsitiktinių poveikių $\epsilon(t)$.
Jei klientas i naudoja AR modelį, tai

$$z_i(t+1) = \sum_{k=1}^{p_i} a_i^k \ z_{t-k} + \epsilon_i(t+1). \qquad (62)$$

Čia parametrai a_k nustatomi
mažiausių kvadratų metodu

$$\min_{a_i} \sum_{s=t_0}^{t} \ \epsilon_i^2(s), \qquad (63)$$

kur

$$\epsilon_i(s) = z(s) - \sum_{k=1}^{p_i} \ a_i^k \ z(s-k). \qquad (64)$$

Šis modelis generuoja laiko eilutes skirtas
akcijų kursui modeliuoti.

Akcijų biržos Nash'o modelis

Ieškom tokių (p_1, p_2) kurių keisti neapsimoka. Naudojam mišrias strategijas

$$x_{p_i},\ i = 1, 2,\ p_i = 1, ..., P_i,\ \sum_{p_i=1}^{P_i} x_{p_i} = 1,\ \ (65)$$

$$0 \le x_{p_i} \le 1.$$

kur x_{p_i} tai tikymybė, kad "atsiminsim" p_i "dienų". "Apgavinis" vektorius:

$$x^1_{p_1} = \arg\max_{x_{p_1}} u^K_1(T_0, T, x_{p_1}, x^0_{p_2}), \qquad (66)$$

$$x^1_{p_2} = \arg\max_{x_{p_2}} u^K_1(T_0, T, x^0_{p_1}, x_{p_2}) \qquad (67)$$

kur $x^0_{p_i},\ i = 1, 2$ sutartinis vektorius.

Nash'o pusiausvyra akcijų biržoje

Nash'o pusiausvyra:

$$(x^*_{p_1}, x^*_{p_2}) = \arg \min_{x^0_{p_1}, x^0_{p_2},} ||(x^1_{p_1}, x^1_{p_2}) - (x^0_{p_1}, x^0_{p_2})||^2 \quad (68)$$

Minimizuojam stochastiniais globaliais metodais.
Jei minimumas didelis,
tikrinam pusiausvyros egzistavimo sąlygas tirdami pelno funkcijų iškilumą.
Galutinis tikslas
-nustatyti optimalią AR modelio "atmintį".
Be to, įdomu palyginti ARMA prognozes naudojant biržos modelį bei realius duomenis.

Kintamų mastelių modelis

$$Z_i = z_{p(i)}\ s_{p(i)}. \tag{69}$$

kur

$Z_i = (z_{ij},\ j = 1, ..., m)$ prognozuojamas valandinis grafikas dienai i,

z_{ij} vidutinė reikšmė dienos i valandą j,

$z_{p(i)}$ vidutinė reikšmė diena $p(i)$,

$p(i)$ nurodo tą dieną kuri buvo panaši į dieną i.

Panašūs būna sekmadieniai, šeštadieniai, darbo dienos ir pan.

Todel, kito sekmadienio grafikas bus tas pats, tik mastelis bus kitas. Mastelis

$$s_{p(i)} = \frac{z_{i-1}}{z_{p(i-1)}}. \tag{70}$$

Çia z_{i-1} šios dienos vidurkis,

$z_{p(i-1)}$ panašios dienos vidurkis, kur $p(i-1) < p(i)$.

”Call Center” modelis

Užsakymų generavimas

Tegu
$F_a(t) = P\{\tau < t\}$
tikimybė, kad laikas iki kito užsakymo τ
bus mažesnis už t,
kur
a vidutinis užsakymu skaičius per valandą.

Puasono sraute

$$F_a(t) = 1 - e^{-1/at}, \tag{71}$$
$$\tau = -aln(1 - \xi). \tag{72}$$

kur ξ tolygiai pasiskirstęs intervale [0,1].
Kai aptarnavimo laikas eksponentinis

$$F_x(t) = 1 - e^{-1/xt}, \tag{73}$$
$$\tau = -xln(1 - \xi), \tag{74}$$

kur x serverio galingumas (užs./val.)

Serverių skaičiaus optimizavimas

Puasono atvėju vidutinis laukimo laikas

$$\gamma = \frac{a}{mx(mx-a)}, \tag{75}$$

kur
x vieno serverio galingumas,
m serverių skaičius.
Bendros aptarnavimo išlaidos

$$c(a,m) = c_m m + c_\gamma(a)\gamma \tag{76}$$

kur
c_m išlaidos vieno serverio išlaikymui,
$c_\gamma(a)$ laukimo laiko piniginis įvertinimas.
Optimalus serverių skaičius

$$m(a) = \min_m c_(a,m). \tag{77}$$

Optimizuojant m svarbu prognozuoti a.

Tvarkaraščių optimizavimas

Siuvyklos ("flow-shop") uždavinys

Čia mašinų eilės tvarką nusako technologija,
pavyzdžiui: žirklės, adata, lygintuvas
Operacijų laikai τ_{ij} rodo kiek laiko reikia
atlikti darbą i su mašina j,
pavyzdžiui, τ_{ij} sako
kiek laiko kerpam švarką i su žirklėmis j
Optimizuojam darbų trukme ("make-span")
Jei darbų nedaug, galim peržiūrėti visas
darbų sekas, pavyzdžiui, $2, 1, 4, 6, 5, 3, 7$
Jei daug, naudojam heuristinius metodus, t.y.
taikom ekspertines išrinkimo taisykles,
bei optimizuojam jų tikimybes,
efektyvumui padidinti.

Mokyklos tvarkaraščio optimizavimas

Čia darbai tai mokinių klasės,
mašinos tai disciplinos,
mašinų eilės tvarka laisva,
resursai tai mokytojai ir specialūs kabinetai,
tikslas minimizuoti mokytojų "langus",
mokinių langai neleistini.
Pradinis tvarkaraštis gerinamas
heuristiniais metodais:
jei atsitiktinis skaičius $\xi < x, \ \xi \in [0, 1]$
tai eilinio mokytojo langas keiciamas
į "patogią" pamoką
priešingu atvėju, mokytojas praleidžiamas,
Patogi yra pirma arba paskutinė pamoka.
Praleidimo tikimybė $1 - x$ optimizuojama
efektyvumui padidinti.

Nuoseklūs sprendimai ir dinaminis programavimas

"Vienkartinės" nuotakos uždavinys

Čia nuotaka teka tik vieną sykį.
Galimų jaunikių skaičius N.
Jaunikio gerumas ω.
Gerumų pasiskirstymas Gauso

$$p(\omega) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-1/2(\frac{\omega-\alpha_0}{\sigma_0})^2}, \tag{78}$$

kur α_0 vidutinis gerumas,
σ_0^2 gerumo išsibarstymas.
Nuotakos įspudis s apie jaunikio gerumą taip pat Gauso

$$p(s|\omega) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2(\frac{s-\omega}{\sigma})^2}. \tag{79}$$

kur σ nuotakos stebėjimo paklaida.

Gerumo įvertinimas kai įspudis s

Jaunikių gerumą ω kai įspudis s nusako Bayes'o formulė

$$p(\omega|s) = \frac{p(s|\omega)p(\omega)}{\int_{-\infty}^{\infty} p(s|\omega)p(\omega)d\omega}. \tag{80}$$

Laukiamas jaunkio naudingumas kai įspudis s

$$u(s) = \int_{-\infty}^{\infty} \omega p(\omega|s)d\omega, \tag{81}$$

kur $p(\omega)$ tikmybinis tankis.
Kai gerumai bei įspudžiai diskretūs, integralas virsta suma

$$u(s_j) = \sum_i \omega_i P(\omega_i|s_j), \tag{82}$$

kur
$P(\omega_i|s_j)$ tikimybė, kad gerumas $\omega = \omega_i$, kai jaunikis daro įspudį $s = s_j$.

Nuotakos sprendimo funkcija

Laukiamas naudingumas kai peršasi paskutinis N-tasis jaunikis

$$u_N(s) = u(s), \tag{83}$$

nes tekėti būtina pagal uždavinio sąlygas. Optimalus sprendimas $d_{N-1}(s)$, kai peršasi $(N-1)$-as jaunikis, randamas pagal formules:

$$u_{N-1}(s) = \max_d (du(s) + (1-d)u_N), \tag{84}$$
$$d_{N-1}(s) = arg \max_d (du(s) + (1-d)u_N). \tag{85}$$

Analogiskai randam optimalų sprendimą kai peršasi $(N-n)$-tas jaunikis.
Taip gaunam nuotakos sprendimo funkciją kuri nurodo kritinio įspudžio s priklausomybę nuo jaunikio eilės numerio n.

”Daugkartinės” nuotakos sprendimo funkcija

Nuotaka gali tekėti ir skirtis N kartų.
Jaunikio gerumą žymėsim ω.
Skaitysim, kad nuotaka ”aiškairegė”,
t.y $\omega = s$.
Turimo vyro gerumą žymėsim q.
Atsisakymo tekėti kainą žymėsim $c = \tau - l$,
kur
τ laukimo nuostoliai,
l skirybų kaina.
Tada vidutinis naudingumas kai peršasi paskutinis N-tasis jaunikis

$$u_N(\omega, q) = \max_d (d\omega + (1-d)(q_N - c_N)). \quad (86)$$

Optimalus sprendimas
priklauso nuo dviejų kintamujų,
vyro gerumo q_N ir jaunikio gerumo ω_N:

$$d_N^* = \begin{cases} 1, & \text{if } \omega_N > q_N - c_N, \\ 0, & \text{if } \omega_N \le q_N - c_N. \end{cases} \quad (87)$$

Maksimalus naudingumas, kai peršasi $(N-i)$-as jaunikis

Maksimalus naudingumas priklauso nuo dvieju parametrų, q ir ω.

$$u_{N-i}(\omega, q) =$$

$$\max_d (d\omega + (1-d)(u_{N-i+1}(q_{N-i+1}) - c_{N-i})).$$

Čia $u_{N-i+1}(q)$ yra laukiamas naudingumas $(N-i+1)$-mais metais, kai turimo vyro gerumas yra q.

$$u_{N-i+1}(q) =$$

$$\int_{-\infty}^{\infty} u_{N-i+1}(\omega, q) p_{N-i+1}(\omega) d\omega. \qquad (88)$$

Sprendimo funkcija kai peršasi $(N-i)$-as jaunikis

$p_{N-i+1}(\omega)$ tai gerumo tikimybių pasiskirstymas $(N-i+1)$-mais metais.

$$q_{N-i+1} = \begin{cases} \omega_{N-i}, & \text{if } q_{N-i+1} < q^*_{N-i}, \\ q_{N-i}, & \text{if } q_{N-i+1} \geq q^*_{N-i}. \end{cases}$$

q^*_{N-i} gaunam iš lygties

$$\omega_{N-i} = u_{N-i+1}(q^*_{N-i}) - c_{N-i}. \tag{89}$$

Optimalus sprendinys $(N-i)$-ais metais

$$d^*_{N-i} = \begin{cases} 1, & \text{if } \omega_{N-i} > u_{N-i+1}(q_N - i + 1) - c_{N-i}, \\ 0, & \text{if } \omega_{N-i} \leq u_{N-i+1}(q_{N-i+1}) - c_{N-i}. \end{cases}$$

Čia sprendimo funkcijos d^*_{N-i} priklauso nuo dviejų kintamųjų ω, q.
Tai sunkina skaičiavimus ir grafinį atvaizdavimą

Dietos uždavinys ir tiesinis programavimas (LP)

$$min_x \sum_{i=1}^{m} c_i x_i, \tag{90}$$

su sąlyga, kad

$$\sum_{i=1}^{m} a_{ij} x_i \geq b_j, \ \ j = 1, ..., n \tag{91}$$

$$x_i \geq 0, \imath = 1, ..., m \tag{92}$$

Pavyzdžiui,
c_1 kilogramo duonos kaina,
a_{11} kaloriju kiekis kilograme duonos,
b_1 reikalingas kalorijų kiekis.
Naudojami simplekso arba
vidinio taško metodai.

Simplekso algoritmas

Pavyzdys:

$$min_x z \tag{93}$$

$$z = x_1 - x_3 \tag{94}$$

$$x_1 + x_2 + x_3 = 1, x_i \geq 0 \tag{95}$$

Čia x_2 bazinis kintamasis,

x_1, x_3 laisvi kintamieji. $x_1 = 1$ yra bazinis sprendinys gautas kai

laisvi kintamieji $x_1 = x_3 = 0$, tada $z = 0$.

Ši sprendinį galima pagerinti didinant x_3, nes $c_3 = -1 < 0$.

Kitas bazinis sprendinys

$x_3 = 1, x_1 = x_2 = 0$, kur $z = -1$, bus optimalus,

nes abiejų laisvų kintamujų c_i neneigiami: $c_1 = 1, c_2 = 0$,

todel didinant x_1, x_2 nesumažinsi z.

Vagies uždavinys ir sveikaskaitinis programavimas (ILP)

$$\max_x \sum_{i=1}^{m} c_i x_i, \tag{96}$$

su sąlyga, kad

$$\sum_{i=1}^{m} g_i x_i \leq g, \tag{97}$$

$$x_i = \{0, 1\}. \tag{98}$$

Çia
c_i daikto i kaina,
g_i daikto i svoris,
g leistinas svoris.
$x_i = 1$ daikta i imti, $x_i = 0$, neimti
Kai m nedidelis,
naudojami šakų ir ribų metodai.
Kai m didelis,
naudojami heuristiniai metodai.

Šakų ir ribų metodas

Pavyzdys- vagies uždavinys:

$$\max_x \sum_{i=1}^{m} c_i x_i \tag{99}$$

$$\sum_{i=1}^{m} g_i x_i \le g,\ \ x_i = \{0, 1\}. \tag{100}$$

Ribos gaunamos iš pagalbinių LP uždavinių:

$$C_1 = c_1 + \max_x \sum_{i=2}^{m} c_i x_i \tag{101}$$

$$g_1 + \sum_{i=1}^{m} g_i x_i \le g \tag{102}$$

$$C_0 = \max_x \sum_{i=2}^{m} c_i x_i \tag{103}$$

$$\sum_{i=2}^{m} g_i x_i \le g \tag{104}$$

"Kertam" šaką 0, jei $C_0 < c_1$.
Šaka 0 reiškia neimti 1-jo daikto,
šaka 1- imti.

Nepalankiausias atvėjas

Nepalankiusiu atvėju, kai
$c_i = c,\ g_i = g_i,\ i = 1, ..., m$
nenukirsim nei vienos šakos
ir todėl, teks sulyginti visas
$N = 2^m$
kombinacijų.
Atvėju, kai
$h_i = c_i / g_i = h,\ i = 1, ..., m$
galim nukirsti vieną-kitą šaką,
tačiau vis vien teks sulyginti beveik visas
$N = 2^m$
kombinacijas.
Palankiu atvėju, kai h_i skiriasi,
nukirsim daug šakų.

Heuristiniai metodai

Pavyzdys- vagies uždavinys:

$$\max_x \sum_{i=1}^{m} c_i x_i, \qquad (105)$$

$$\sum_{i=1}^{m} g_i x_i \leq g,\ x_i = \{0,1\}. \qquad (106)$$

Çia heuristika

$$h_i = c_i / g_i. \qquad (107)$$

"Gryna" heuristika kai imam daiktą

$$i^* = \arg\max_i h_i. \qquad (108)$$

Čia $N \leq m^2$ tačiau galim "įstrigti".
Todel naudojam "mišrias" heuristikas.

Mišrios heuristikos

Pavyzdys- tas pats vagies uždavinys:
"Mišri" heuristika kai imam daikta i
su tikimybe r_i.
Tradicinis būdas

$$r_i = h_i / \sum_j h_j. \tag{109}$$

Bayes'o heuristinio algoritmo iliustracija:

$$r_i^0 = 1/m, \tag{110}$$

$$r_i^1 = h_i / \sum_j h_j, \tag{111}$$

$$r_i^\infty = 1,\ jei\ h_i = \max_j h_j. \tag{112}$$

Heuristikų ”mišinio” optimizavimas

Trijų heuristikų ”mišinį”
parenkam pagal ”loteriją”
$x = (x_0, x_1, x_2)$,
kur
x_0 tikimybė naudoti algoritmą (110),
x_1 tikimybė naudoti algoritmą (111),
x_2 tikimybė naudoti algoritmą (112).
Loteriją x optimizuojam Bayes'o metodu
taip, kad atlikus K iteracijų
gauti geriausią rezultatą.
Todėl tai vadinam Bayes'o heuristiniu metodu.
Kitas šio metodo pavyzdys
tai mokyklos tvarkaraštis, kur x tikimybė, kad eilinio mokytojo ”langai” bus naikinami šioje iteracijoje.
Trečias pavyzdys- siuvyklos uždavinys
čia h_i tai darbo i vykdymo laikas.

Optimizavimo sudėtingumas

Algoritmas polinominis, kai skaičiavimo laikas $T = Cm^K$.
Čia
K sveikas skaičius,
m uždavionio sudėtingumas.
Diskretiniu atvėju, m kintamųjų skaičius.
Tolydiniu atvėju, m garantuotas tikslumas,
kai maksimali paklaida $\epsilon \leq 2^{-m}$.
Algoritmas eksponentinis, kai skaičiavimo laikas $T \geq C2^m$.
"Ribiniu" atvėju tai NP-pilna klasė.
Ši klasė pasižymi tuo, kad nėra įrodyta ar čia
būtinai reikia eksponentinio algoritmo,
ar užtenka ir polinominio.
Polinominio algoritmo NP-pilnai klasei
dar niekas nesugalvojo,
tačiau neįrodyta, kad jo nėra.

Sudėtingumo pavyzdžiai

Uždavinių sudėtingumo pavyzdžiai:
Tiesinio programavimo- polinominiai.
Vagies ir siuvyklos - NP-pilni.
Daugelio kintamųjų globalinio optimizavimo bendru atvėju - eksponentiniai.
Čia skaičiavimo laikas
$T \geq C2^{mn}$,
kur m tikslumas, n kintamųjų skaičius.

Lokalus optimizavimas

Nusileidimo metodai

Tikslo funkcija $f(x)$, $\quad x = (x_i,\ i = 1, ..., m)$,

$$x^{n+1} = x^n - \alpha_n s_n. \tag{113}$$

Žingsnio ilgis

$$\alpha_n = \arg min_\alpha f(x^n - \alpha_n s_n). \tag{114}$$

Čia s_n žingsnio kryptis, n žingsnio numeris. Metodas gradientinis, kai

$$s_n = grad\ f(x^n), \tag{115}$$

kur gradientas

$$grad\ f(x^n) = (\frac{\partial f(x^n)}{\partial x_i},\ i = 1, ..., m). \tag{116}$$

Newton'o metodai

Metodas Newton'o, kai

$$s_n = H_n^{-1} grad\ f(x^n), \tag{117}$$

kur Hessian'as

$$H_n = (\frac{\partial^2 f(x^n)}{\partial x_i\ \partial x_j},\ i,j = 1,...,m). \tag{118}$$

Metodas kvazi-Newton'o, kai

$$H_n^{-1} \approx G_n, \tag{119}$$

kur

$$G_{n+1} = G_n - \frac{(G_n\gamma_n)(G_n\gamma_n)^T}{\gamma_n^T G_n \gamma_n} + \frac{\delta_n\ \delta_n^T}{\delta_n^T\ \gamma_n}. \tag{120}$$

Čia T transponavimo ženklas,

$$\gamma_n = grad\ f(x_n) - grad\ f(x_{n-1}), \tag{121}$$

$$\delta_n = x^n - x^{n-1}, \tag{122}$$

$$G_0 = I. \tag{123}$$

Lokalus konvergavimas

Gradientinis metodas konverguoja,
kai $f(x)$ diferencijuojama ir iškila.
Newton'o ir Kvazi-Newton'o metodai
konverguoja, kai $f(x)$
du kartus diferencijuojama ir iškila.
Gradientinis metodas konverguoja lėtai

$$||x^n - x^*|| \to \ 0. \tag{124}$$

Newton'o ir kvazi-Newton'o metodai
konverguoja greitai

$$\frac{||x^n - x^*||}{||x^{n-1} - x^*||} \to \ 0, \tag{125}$$

kur x^* optimumas.
Newton'o metodo trukumas-
reikia surasti atvitkštinį Hessianą H_n^{-1}. Todėl,
Newton'o metodas nedirba kai $\det H_n = 0$.
Šio trukumo neturi kvazi-Newton'o metodas.

Optimizavimas su ribojimais

$$\max_x \ f_0(x), \tag{126}$$

$$f_j(x) \leq c_j, \ j = 1, ..., n. \tag{127}$$

Baudų metodas

$$\max_x \{f_0(x) - \sum_j b_j (f_j(x) - c_j)^2\}, \tag{128}$$

kur baudos

$$b_j \ = \ \begin{cases} b, & \text{jei } f_j(x) > 0, \\ 0, & \text{jei } f_j(x) \leq 0. \end{cases} \tag{129}$$

Trukumas-neaiskus baudos b dydis:
kai b mažas- bus pažeisti ribojimai,
kai b didelis- gali būti perpildymų.

Lagranžo daugikliai

Uždavinys (126)(127) keičiamas uždaviniu

$$\min_{\lambda > 0} \max_x \{f_0(x) - \sum_j \lambda_j \ (f_j(x) - c_j)\}. \quad (130)$$

Ekonominė interpretacija:
$f(x)$ pelnas,
x technologija,
c_j resursas j,
λ_j resurso j kaina,
sąlyga $\min_{\lambda > 0}$ reiškia, kad resurso c_j savininkas nustato kainas λ_j tokias kurios maksimizuoja jo pelną,
sąlyga $\max_x$ reiškia, kad įmonės $f(x)$ savininkas renkasi tokią technologiją x kuri maksimizuoja jo pelną, esant kainoms λ.

Globalus optimizavimas

Tolygus tinklelis

$$r_n = \max_x \min_i ||x - x^i||. \tag{131}$$

Tinklelis $x(n) = (x^i,\ i = 1, ..., n)$ tuo geresnis, kuo didžiausia "skylė" r_n mažesnė

$$R_n = arg \min_{x(n)} \ r_n. \tag{132}$$

Toks tinklelis minimizuoja maksimalią paklaidą ϵ, optimizuojant Lifšico funkcijas

$$\frac{||f(x^i) - f(x_j)||}{||x^i - x_j||} \leq L < \infty, \tag{133}$$

kur L Lifšico konstantė,
$\epsilon \leq LR_n$.
Kai x turi daug komponenčių,
tolygų tinklelį sunku sudaryti.
Tada naudojam aproksimacijas:
arba geresnę "Lptau",
arba primityvią "Monte Carlo", kai x^i parenkami su vienodomis tikimybėmis.

Bayes'o metodai

Tolygūs tinkleliai minimizavo
maksimalią paklaidą,
todel reikalavo
$T = C2^{mn}$
laiko,
kur n kintamųjų skaičius,
m nurodo maksimalią paklaidą:
$\epsilon \leq 2^{-m}$.
Tai labai brangu, kai m arba n dideli.
Tada naudojam Bayes'o metodus
kurie minimizuoja vidutinę paklaidą,
duotam iteracijų skaičiui.
Tam reikia parinkti statistinį funkcijos $f(x)$ modelį.
Kai $n = 1$ patogus Wiener'io modelis.

Wiener'io modelio optimizavimas

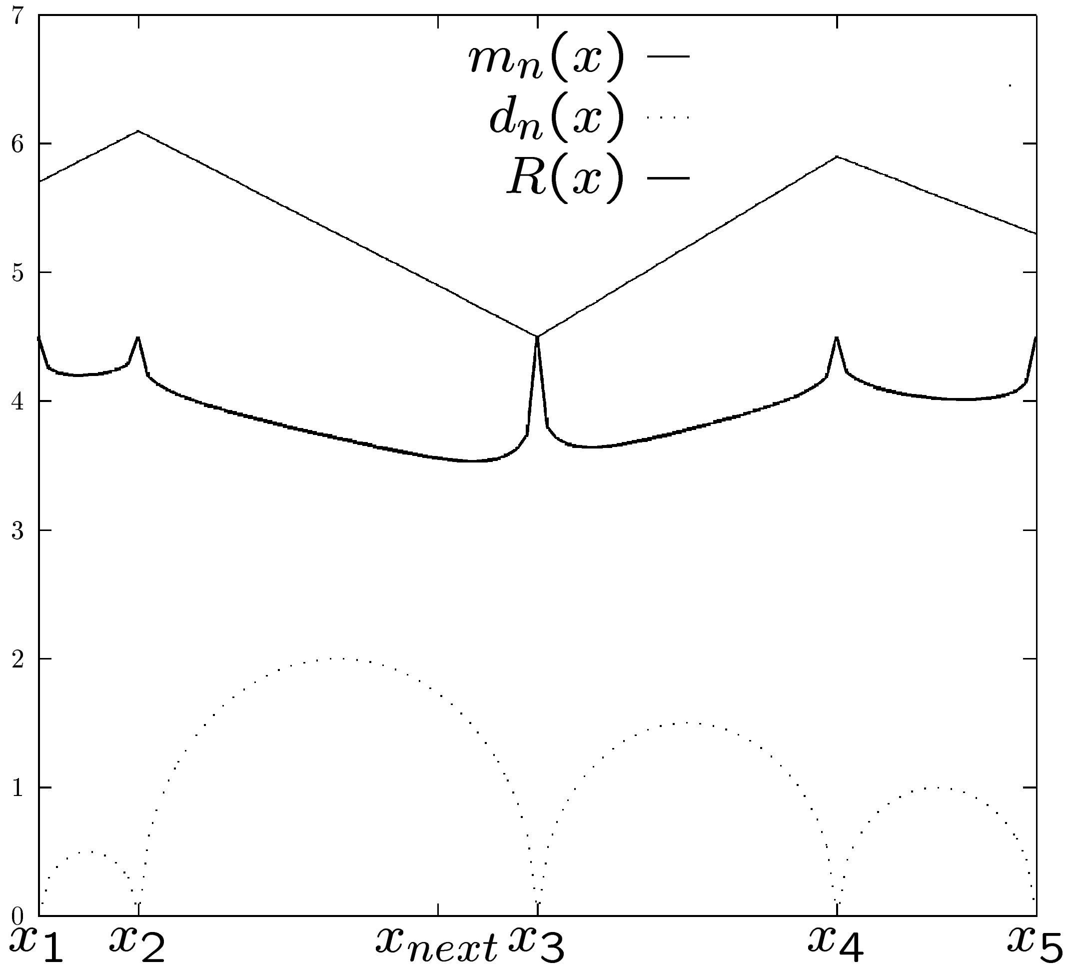


Figure 1: Wiener'io modelis

Sąlyginis vidurkis $m_n(x)$,
Sąlyginė dispersija $d_n(x)$,
Rizikos funkcija $R(x)$,
Bayes'o metodas:

$$x^{n+1} = arg\min_x R(x). \tag{134}$$

Rizikos funkcija

Wiener'io modelio rizikos funkcija

$$R(x) = \frac{1}{\sqrt{(2\pi d_n(x))}} \int_{-\infty}^{+\infty} \min\ (c_n, y_x)\ e^{-\frac{(y_x - m_n(x))^2}{d_n(x)}} dy_x$$

Čia
$y_x = f(x)$,
$c_n = \min_i f(x^i) - \epsilon,\ \ \epsilon > 0$.
Matom, kad
$R(x) = c_n$, jei $x = x^i$
nes
$d_n(x^i) = 0$.

Koordinatinis optimizavimas

Taikant Wiener'io modelį kai $m > 1$, patogus koordinatinis otimizavimas

$$x^1 = arg \min_{x_1} f(x^0) \quad (135)$$

$$x^2 = arg \min_{x_2} f(x^1) \quad (136)$$

....

$$x^m = arg \min_{x_m} f(x^{m-1}) \quad (137)$$

....

Trukumas:
rezultatas priklauso nuo x^0.
Privalumas:
patogi vizualizacija,
nes projekcijos vaizdžios,
kadangi kintamieji keičiami po vieną.
Metodas konverguoja, jei

$$f(x) = \sum_i f_i(x_i). \quad (138)$$

Daugiamatis Bayes'o metodas

$$x^{n+1} = arg \min_x R(x) \qquad (139)$$

$$R(x) = y_{0n} - \min_i \frac{||x - x_i||^2}{f(x_i) - c_n}, \qquad (140)$$

$$c_n = \min_i f(x^i) - \epsilon. \qquad (141)$$

Kai n didelis

$$d^*/d_a = (\frac{f_a - f^* + \epsilon}{\epsilon})^{1/2}$$

Čia
d^* stebėjimų tankis,
f^* funkcijos $f(x)$ vidurkis
(abu globalaus optimumo x^* aplinkoje)
d_A vidutinis stebėjimų tankis,
f_A funkcijos $f(x)$ vidurkis,
(abu visoje srityje A).

Vektorinis optimizavimas

Maksimizuojam vektorinę tikslo funkcija

$$f(x) = (f_i(x),\ i = 1, ..., m). \tag{142}$$

Pareto optimumas tai aibė X^*.
$x^* \in X^*$, jei neatsiras tokio x, kad

$$f_i(x) \geq f_i(x^*),\ \forall i \tag{143}$$

$$f_j(x) > f_j(x^*),\ \exists j \tag{144}$$

Skaliarizacija:

$$x(c) = arg\max_x \sum_i c_i f_i(x) \tag{145}$$

$$c_i > 0 \tag{146}$$

Čia $x(c) \in X^*$
Ribojimai:

$$x(b) = arg\max_x f_1(x) \tag{147}$$

$$f_i(x) \geq b_i,\ i = 2, ..., m \tag{148}$$

Čia $x(b) \in X^*$, jei $x(b)$ vienintėlis.

Simulated Annealing (SA)

SA yra labai paplitęs
globalinio optimizavimo metodas. Žymėsim

$$h_j = f(x^j) - f(x^{j-1}), \tag{149}$$

jei $h_j \geq 0$, tai parenkamas x^j
jei $h_j < 0$, tai x^j parenkamas su tikimybe

$$r_j = \begin{cases} e^{\frac{h_j}{x/\ln(1+N)}}, & \text{if } h_j < 0, \\ 1, & \text{otherwise.} \end{cases} \tag{150}$$

SA mėgstamas dėl jo paprasumo
ir dėl gero teorinio pagrindimo
SA praktini efektyvumą galima padidinti,
optimizuojant x, pavyzdžiui, Bayes'o metodu

Lentpjuvės uždavinys

Reikia minimizuoti atliekas
išpjaunant $i = 1, ..., m$ lentų,
turint $j = 1, ..., n$ rastų.
Visų jų dydžiai žinomi.
Žymėsim h_{ij} prioriteto taisyklę
siūlančią pjauti lentą i is rasto j.

Pagrindinis sunkumas
tai sugalvoti gerą heuristiką h_{ij}.
Toliau kaip vagies uždaviny (93)-(100).
Pagrindiniai skirtumai:
- ten heuristikos h_i daiktams, kurių
vertę maksimizuojam, laikantis svorio ribojimo,
- čia heuristikos h_{ij} poroms lenta-rastas,
minimizuojam atliekas laikantis ribojimo,
kad lentos tilptų rastuose.

Lentpjuvės uždavinio formulavimas

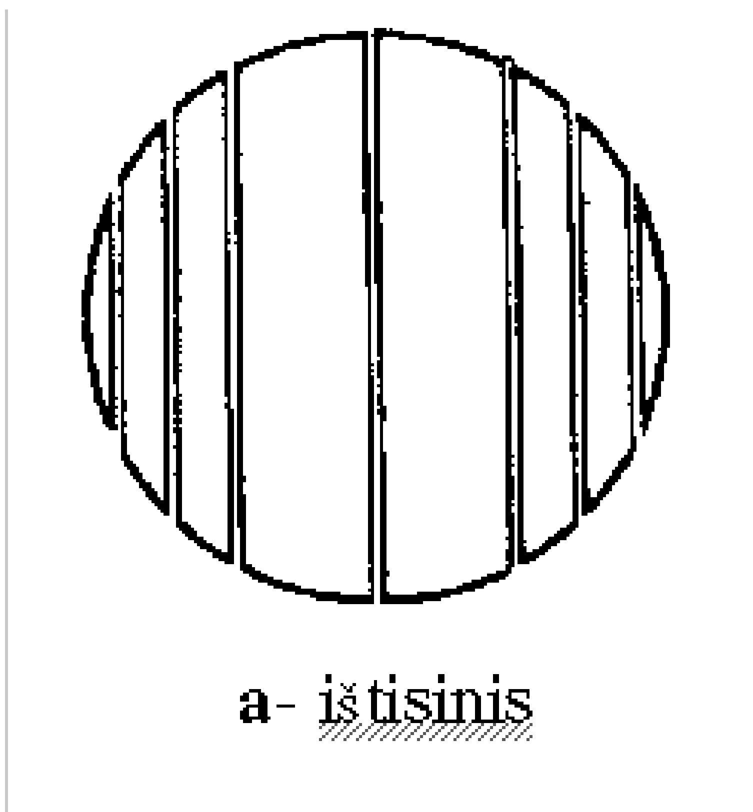
$$\min_y \ v(y), \ y = (y_{ijk}), \quad (151)$$
$$i = 1, ..., m, \ j = 1, ..., n, \ k = 1, ..., K,$$
$$a_i y_{ijk} \le d_{ijk}.$$

Čia $v(y)$ yra atliekos, jos nustatomos kaip skerspjuvių skirtumas tarp išpjautyų lentų ir panaudotų rastų.
a_i yra lentos i storis.
d_{ijk} yra rasto j storis ten kur lenta i padėta vietoje k,
žiur. figura "Ištisinis sastatas".

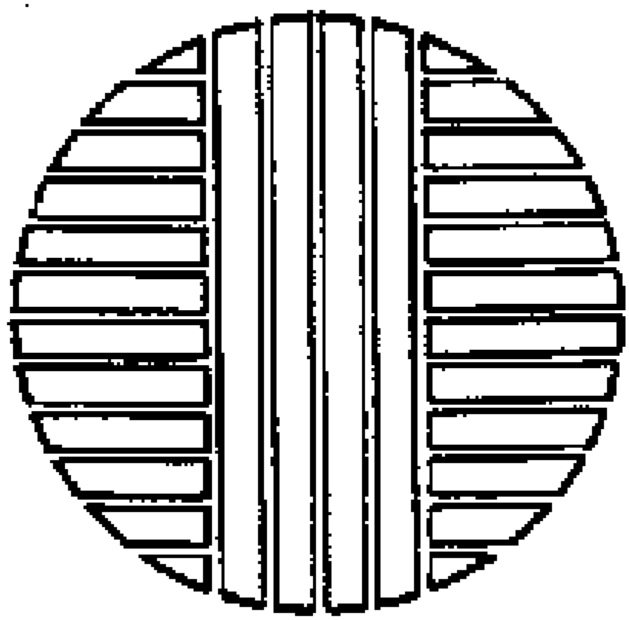
$$y_{ijk} = \begin{cases} 1, & \text{jei } i \in (j,k), \\ 0, & \text{jei ne.} \end{cases}, \quad (152)$$

kur i yra lentos numeris.
j yra rasto numeris.
k nurodo i lentos vietą raste j.
$i \in (j,k)$ reiškia, kad lenta i yra raste j vietoje k.

Ištisinis sąstatas



Segmentinis sąstatas



c - ištisinis segmentinis

Traukinių formavimas

Minimizuoti traukinių skaičių $n = n(y)$, tikslu išvežti m vagonų iš prekių stoties.

$$\min_y \; n(y), \tag{153}$$

$$\sum_{i=1}^{m} a_i y_{ij} \le A_j, \tag{154}$$

$$\sum_{i=1}^{m} b_i y_{ij} \le B_j. \tag{155}$$

Čia

$$y_{ij} = \begin{cases} 1, & \text{jei } i \in j, \\ 0, & \text{jei ne.} \end{cases} . \tag{156}$$

a_i yra vagono i svoris.
A_j yra leistinas traukinio j svoris.
b_i yra vagono i ilgis.
B_j yra leistinas traukinio j ilgis.
$i \in j$ reiškia, kad vagonas i yra traukiny j.

LOŠIMŲ TEORIJA IR TAIKYMAI

Tai tęsinys optimizavimo metodų kurso. Optimizavimo kurse lošimų teorijos uždavinius nagrinėjom be teorinių pagrindų, tik kaip optimizavimo taikymo pavyzdžius.

Lošimų teorijos kurse bus teorija šiems uždaviniams pagristi.
Be to bus naujų lošimų bei rinkos teorijos uždavinių.

Namų darbai bus arba tobulinimas optimizavimo pavyzdžių,
arba sprendimas naujų uždavinių.

Paprasčiausias lošimas

Tai du lošėjai, du ėjimai.
Pirmojo išlošimas

$$u(i,j) = \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \tag{157}$$

kur $i = 1, 2$ yra 1-jo ėjimas,
$j = 1, 2$ yra 2-jo ėjimas.
Antrojo išlošimas priesingas

$$v(i,j) = \begin{vmatrix} -1 & 1 \\ 1 & -1 \end{vmatrix} \tag{158}$$

Tai antagonistinis lošimas:
ką vienas išlošia, kitas pralošia.
Lošimui aprašyti užtenka
vienos matricos $u(i,j)$.
Todėl tai matricinis lošimas.

Grynos ir mišrios strategijos

Gryna strategija, kai ėjimai daromi nenaudojant randomizacijos (lietuviškai "burtų").

Mišri strategija, kai ėjimai daromi burtu keliu, pagal lošėjų nustatytas tikimybes.
Čia svarbu vidutinis išlošimas

$$U(x,y) = x_1y_1 - x_1y_2 - x_2y_1 + x_2y_2$$
$$V(x,y) = -x_1y_1 + x_1y_2 + x_2y_1 - x_2y_2 \quad (159)$$

kur
$0 \le x_i \le 1,\ i = 1,2$ tai tikimybės, kad 1-sis lošėjas darys ėjimą i,
analogiškai y_i
Iš čia

$$U(x,y) = 4x_1y_1 - 2x_1 - 2y_1 + 1$$
$$V(x,y) = -4x_1y_1 + 2x_1 + 2y_1 - 1 \quad (160)$$

Pusiausvyros strategijos

Tai tokios strategijos,
kurių neapsimoka keisti.
Grynų strategijų aibėje to nėra.

Vieninelė pusiausvyros strategija mišri

$$x_i = y_i = 0.5 \tag{161}$$

$$U(x_1 = 1/2, y_1 = 1/2) = 0,$$
$$V(x_1 = 1/2, y_1 = 1/2) = 0 \tag{162}$$
$$x_2 = 1 - x_1, y_2 = 1 - y_1$$

nes čia

$$U(x_1 = 1/2 + \epsilon, y_1 = 1/2) < 0,$$
$$V(x_1 = 1/2, y_1 = 1/2 + \epsilon) < 0, \tag{163}$$

jei $\epsilon \neq 0$.

Bimatricinis lošimas

Čia $u(i,j) \neq -v(i,j)$, pavyzdžiui

$$u(i,j) = \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \tag{164}$$

$$v(i,j) = \begin{vmatrix} -1 & 1 \\ 1 & 0 \end{vmatrix} \tag{165}$$

Bimatricinio lošimo strategijos

Vidutinis išlošimas

$$U(x,y) = x_1y_1 - x_1y_2 - x_2y_1 + x_2y_2,$$
$$V(x,y) = -x_1y_1 + x_1y_2 + x_2y_1, \quad (166)$$

arba

$$U(x,y) = 4x_1y_1 - 2x_1 - 2y_1 + 1,$$
$$V(x,y) = -3x_1y_1 + x_1 + y_1. \quad (167)$$

Čia pusiausvyros strategijos

$$y_1 = y_2 = 1/2, \quad (168)$$
$$x_1 = 1/3, x_2 = 2/3, \quad (169)$$
$$u^* = U(x_1 = 1/3, y_1 = 1/2) = 0, \quad (170)$$
$$v^* = V(x_1 = 1/3, y_1 = 1/2) = 1/3, \quad (171)$$

nes

$$U(x_1 = 1/3 + \epsilon, y_1 = 1/2) < 0, \quad (172)$$
$$V(x_1 = 1/3, y_1 = 1/2 + \epsilon) < 0, \quad (173)$$

jei $\epsilon \neq 0$.

Nejautrus apgavystei algoritmas

Žymesim

$$U(1,y) = y_1 - y_2, \tag{174}$$
$$U(2,y) = -y_1 + y_2, \tag{175}$$
$$V(x,1) = -x_1 + x_2, \tag{176}$$
$$V(x,2) = x_1, \tag{177}$$

kur $U(1,y)$ tai vidutinis 1-jo išlošimas naudojant ėjimą i=1, kai 2-sis eina pagal tikimybes y, kitoms eilutėms panašiai. Strategijos x, y nejautrios apgavystei, jei

$$U(1,y) = U, \tag{178}$$
$$U(2,y) = U,$$
$$V(x,1) = V,$$
$$V(x,2) = V.$$
$$\tag{179}$$

Kadangi čia yra nelygybių $0 \leq x_i \leq 1, 0 \leq y_i \leq 1, x_1 + x_2 = 1, y_1 + y_2 = 1$, tai patogu naudoti tiesinį programavimą, smulkiau aprašytą inspektoriaus uždavinyje.

Grynų strategijų pusiausvyra

Jei $V(x,1) < V(x,2)$, nepriklausomai nuo tikimybių x, tai 2-sis naudos gryną strategiją $j = 2$.
Pavyzdys:

$$u(i,j) = \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \tag{180}$$

$$v(i,j) = \begin{vmatrix} -1 & 1 \\ 1 & 2 \end{vmatrix} \tag{181}$$

Pusiausvyra bus kai $i = 2, j = 2$, tada išlošimai $u(2,2) = 1, v(2,2) = 2$.
Tiesinio programavimo uždavinys (24) šiuo atvėju sprendinio neturi. Todėl, algoritmą (24) reikia papildyti, ieškant grynų strategijų tenkinančių pusiausvyros sąlygas. Tai daroma peržiūrint visas grynas strategijas. Inspektoriaus uždaviny (20) (21) to nereikia.

Sandėris

Jei lošimas neantagonistinis $v(i,j) \neq -u(i,j)$, tai galimas sandėris.
Pavyzdžiui, inspektorius ir brakonierius gali susitarti pasidalinti briedį.
Tačiau, neaišku kaip pasidalinti.
Atsakymą duoda Nash'o sandėris.

Sandėris priklauso nuo leistinų dalybų aibės D, bei nuo to ka lošėjai gautų, jei nesusitartų

$$U^* = max_x \ min_y U(x,y),$$
$$V^* = max_x \ min_y V(x,y). \tag{182}$$

Čia
U^* 1-jo lošėjo maksimalus garantuotas išlošimas, o V^* - 2-jo.
Sudarius sandėrį $(\bar{u}, \bar{v})$, 1-sis gauna $\bar{u}$, 2-sis $\bar{v}$.
Sandėris $(\bar{u}, \bar{v})$ nustatomas vienareikšmiai, jei tenkinamos kelios gana priimtinos sąlygos.

Nash'o aksiomos

1. $(\bar{u}, \bar{v})$,

2. $(\bar{u}, \bar{v}) \in D$,

3. jei $(u, v) \in D$ ir $(u, v) \geq (\bar{u}, \bar{v})$, tai $(u, v) = (\bar{u}, \bar{v})$,

4. jei $(\bar{u}, \bar{v}) \in T \subset D$ ir $(\bar{u}, \bar{v}) = \phi(D, u^*, v^*)$, tai $(\bar{u}, \bar{v}) = \phi(T, u^*, v^*)$,

5. tegu aibę D' gaunam iš D taip $u' = a_1 u + b_1$, $v' = a_2 v + b_2$, tada, iš $\phi(D, u^*, v^*) = (\bar{u}, \bar{v})$ seks, kad $\phi(D', a_1 u + b_1,\ a_2 v + b_2) = (a_1 u + b,\ a_2 v + b_2)$,

6. jei $(u, v) \in D \Leftrightarrow (v, u) \in D$, $u^* = v^*$ ir $\phi(D, u^*, v^*) = (\bar{u}, \bar{v})$, tai $\bar{u} = \bar{v}$.

Sandėrio funkcija

Galiojant Nash'o aksiomoms
egzistuos vienintelė sandėrio funkcija
$\phi(D, u^*, v^*) = (\bar{u}, \bar{v})$

Jei atsiras tokia pora $(u, v) \in D$, kad $u > u^*,\ v > v^*$, tai sandėris

$$(\bar{u}, \bar{v}) = arg\ max_{u,v}(u - u^*)(v - v^*), \quad (183)$$

kur

$$(u, v) \in D,\ u \geq u^*,\ v \geq v^* \quad (184)$$

Parastai bendras išlošimas yra ribotas, tada

$$D = \{(u, v) : u + v \leq c\}. \quad (185)$$

Čia sandėris

$$(\bar{u}, \bar{v}) = ((c + u^* - v^*)/2,\ (c + v^* - u^*)/2) \quad (186)$$

Sandėrio pavyzdys 1

Tai bimatricinis lošimas (164)(165),
kur $(u^*, v^*) = (0, 1/3)$.
Čia bendras išlošimas $u + v$ neribotas,
todėl, iš (183),(170),(171),
sandėris $(\bar{u}, \bar{v}) = (1, 1)$.

Kai bendras išlošimas ribotas, $u + v \leq c = 1$,
sandėris bus kitas.
Iš (186) sandėris $(\bar{u}, \bar{v}) = (1/3, 2/3)$.

Kai bendras išlošimas dar mažesnis, $c = 1/3$,
tai sanderyje $(\bar{u}, \bar{v}) = (0, 1/3)$
1-sis nieko negaus, todėl jame nedalyvaus,
nes lošdamas gaus nemažiau,
kadangi $u^* = \bar{u}$.

Sandėrio pavyzdys 2, inspektorius

Inspektoriaus sandėris su brakonierium.
Lošimą aprašo (164)(165).
Čia bendras išlošimas $c = 1$.
Iš (34), garantuoti išlošimai
$u^* = 2/9, \ v^* = 5/9$.
Iš (186), sandėris
$(\bar{u}, \bar{v}) = (1/3, 2/3)$.

Šis sandėris įvyks, nes
$(1/3, 2/3) > (2/9, 5/9)$

Sandėrio nebus, jei laukiama bauda
inspektoriui už sandėrį
$B > \bar{u} - u^* = 1/9$.
Čia $B = p_b \ b$, kur p_b tai baudos b tikimybė.
Kai bauda $b = 1$ lygi briedžio kainai,
baudos tikimybė $p_b > 1/9$.

Sandėrio pavyzdys 3, streikas

Įmonės šeimininkų strategija $x \in [0, a]$,
tai mokėti atlyginimą x,
kur a įmonės pajamos.
Darbininkų strategijos

$$y = \begin{cases} 0, & \text{streikuoti} \\ 1, & \text{dirbti.} \end{cases} \tag{187}$$

Šeimininkų išlošimas

$$u(x,y) = \begin{cases} a - x, & \text{jei } y = 1 \\ -x, & \text{jei ne.} \end{cases} \tag{188}$$

Darbininkų išlošimas

$$v(x,y) = \begin{cases} x, & \text{jei } x > 0 \\ -b, & \text{jei } x = 0. \end{cases} \tag{189}$$

Čia garantuoti išlošimai $u^* = 0,\ v^* = -b$,
kur b darbininkų nuostoliai,
kai jie nieko negauna.
Čia bendras išlosimas $c = a$.
Tegu $b < a$, tada iš (186) sandėris

$$(\bar{u}, \bar{v}) = ((a+b)/2,\ (a-b)/2). \tag{190}$$

Pusiausvyros apibrėžimas

Yra m lošėjų.
Jie naudoja strategijas $y_j,\ j = 1, ..., m$.
Lošėjo i vidutinis išlošimas
$u_i = u_i(y_j,\ j = 1, ..., m),\ i = 1, ..., m$
priklauso ir nuo to ką daro kiti.
Pradinę srategiją žymėsim
$y^0 = (y_j^0,\ j = 1, ..., m)$. Vadinsim ją sutartine.
Skaitysim, kad individualūs lošėjai pakeis savo sutartines strategijas, jei tai padidins jų išlošimą.
Taip gausim "apgavines" strategijas

$$y_j^1 = arg\ max_{y_j} u_j(y_1^0, ..., y_j, ..., y_m^0), \quad j = 1..., m \tag{191}$$

Nash'o pusiausvyra tai tokia strategija, kurios neapsimokės keisti nei vienam lošėjui.
Taip bus jei

$$y_j^1 = y_j^0,\ j = 1, ...m \tag{192}$$

Kelių lošėjų kooperavimą aptarsim vėliau, kalbėdami apie koalicijas.

Pusiausvyros sąlygos

Pusiausvyra nevisada egzistuoja.
Pusiausvyra egzistuos, jei islošimo funkcijos $U(x,y), V(x,y)$ ir strategijų aibės $x \in X, y \in Y$ bus iškilos.
Funkcija $f(x)$ bus iškila, jei tiesės jungiančios poras $f(x^1), f(x^2) \in X$ bus žemiau $f(x), x \in X$.
Aibė X bus iškila, jei tiesės jungiančios taškų poras x^1, x^2 priklausys aibei X.
Funkcijos iškiluma iliustruoja Figura 1 "Globalus ir lokalus minimumai"
(minimizuojant funkcijos iškilumas dažnai suprantamas "atvirkščiai").
Paprasto lošimo pvyzdy grynų strategijų aibė neiškila, todėl ir pusiausvyros nėra. Mišrių strategijų aibė iškila, tai užtikrina pusiausvyros egzistavimą.
Iliustracijos Walras'o uždaviny.

Kooperatinai lošimai

Kai lošėjų skaičius $m > 2$,
svarbią rolę įgyja koalicijos.
Ypač reikšmingas jų stabilumas.
Trijų serverių pavyzdys yra skaidrėje
"Stabilios koalicijos".
Programiškai jis dar nerealizuotas.

Koalicjų stabilumą apsprendžia tai
ar daugiau gausim lošdami kartu, ar atskirai.
O tai ka gausim, priklauso
ir nuo bendro koalicijos išlošimo
ir nuo šio išlošimo dalybų.

Garantuotą koalicijos išlošimą nurodo
charakteringa funkcija.
Teorinis dalybų pagrindas tai dominavimo sąvoka.

Charakteringa funkcija

S tai visų lošėjų aibė.
$s \subset S$ tai koalicija.
Maksimalus garantuotas koalicijos s išlošimas

$$v(s) = max_x \; min_y \; u_s(x, y). \tag{193}$$

Čia $x = x(s)$, koalicijos s strategija,
$y = y(S \setminus s)$, likusių lošėjų koalicijos strategija,
$v(s)$, charakteringa funkcija.

Lošimas v su pastovia suma, jei

$$v(s) + v(S \setminus s) = v(S). \tag{194}$$

Lošimas esminis, jei

$$\sum_{i \in S} v(i) < v(S). \tag{195}$$

Kooperatinio lošimo pavyzdys 1, trys berniukai

Čia trys lošėjai $i = 1, 2, 3$. Indeksai j žymi koalicijas s_j, kur $s_1 = \{1, 2\}$, $s_2 = \{1, 3\}$, $s_3 = \{2, 3\}$. Koalicijų s_j išlošimai bei jų dalybos lošėjams i

$$u(i, j) = \begin{vmatrix} 1 & 1 & -2 \\ 1 & -2 & 1 \\ -2 & 1 & 1 \end{vmatrix} \tag{196}$$

Lošimo charakteringa funkcija

$$v(s) \;=\; \begin{cases} 2, & \text{jei } |s| = 2 \\ -2, & \text{jei } |s| = 1 \end{cases}. \tag{197}$$

Čia $|s|$ koalicijos narių skaičius. Lošimas v su pastovia suma, nes

$$v(s) + v(S \setminus s) = v(S) = 0. \tag{198}$$

Lošimas esminis, nes

$$\sum_{i \in S} v(i) = -6 < v(S) = 0. \tag{199}$$

Kooperatinio lošimo pavyzdys 2, UAB

Čia keturi akcininkai $i = 1, 2, 3, 4$ ir visos jų koalicijos, po vieną, po du, po tris ir visi keturi kartu. Akcijų kiekis g_i atitinkamai 10, 20, 30, 40. Koalicija s_j išlošia, jei $G_j > 50$, kur $G_j = \sum_{i \in s_j} g_i$. Lošimo charakteringa funkcija

$$v(s_j) \;=\; \begin{cases} 1, & \text{jei } G_j > 50 \\ 0, & \text{jei } G_j \leq 50\ . \end{cases} \tag{200}$$

Lošimas v su pastovia suma, nes

$$v(s) + v(S \setminus s) = v(S) = 0. \tag{201}$$

Lošimas esminis, nes

$$\sum_{i \in S} v(i) = 0 < v(S) = 1. \tag{202}$$

Dalybų kol kas nenurodom.

Dalybos

z_i tai lošėjo i dalis dalybose $z = (z_1, ..., z_m)$, Dalybos realios jei

$$\sum_{i \in S} z_i = v(S) \tag{203}$$

$$z_i \geq v(i) \tag{204}$$

Dalybos z dominuos dalybas w koalicijoje s

$z \succ_s w$,

jei

$$\sum_{i \in s} z_i \leq v(s) \tag{205}$$

$$z_i > w_i, \; i \in s \tag{206}$$

Jei atsiras koalicija s kuriai galios (205)(206), tai rasysim, kad

$z \succ w$

Branduolys

Lošimo v branduoliu $C(v)$
vadinsime aibe visų nedominuojamų dalybų.
Branduolys užtikrina visu koaliciju stabiluma,
nes niekas nenorės pakeitimu.
Praktiškai tai reiškia
ekonominį ir politinį stabilumą.

Dėja, branduolio nebus, $C(v) = \emptyset$,
kai lošimas esminis su pastovia suma

$$\sum_{i \in S} v(i) < v(S),$$
$$v(s) + v(S \setminus s) = v(S).$$

Pavyzdžiuose 1 ir 2 branduolio nebus.
Nash'o uždaviny branduolys čia gali egzistuoti,
nes tai nėra lošimas su pastovia suma.
Kai bus programa, patikrinsim.

Shapley vektorius

Kai nėra branduolio, dalyboms naudoja Shapley vektorių. Shapley dalybos bus stabilios, jei visi lošėjai supras šias Shapley sąlygas ir su jomis sutiks. Tegu

1. $\sum_{i \in s} \phi_i[s] = v(s)$,
   kur $\phi_i[s]$ Shapley dalybos,

2. $\phi_{\pi(i)}[\pi v] = \phi_i[v]$, kur π lošėjų perstatymas,

3. $\phi_i[u + v] = \phi_i[u] + \phi_i[v]$, kur u ir v du lošimai.

Tada egzistuos vieninelė Shapley dalybų funkcija

$$\phi_i[v] = \sum_{s \subset S,\ i \in s,} (|s| - 1)!(|S| - |s|)!/|S|!$$
$$(v(s) - v(s \setminus \{i\}), \quad (207)$$

Matom, kad "nenulinės" yra tik "įdomios" lošėjui i koalicijo s, t.y. tos kurios su juo išlošia,
o be jo ne. Jų aibes žymėsim S_i.

Shapley dalybų pavyzdžiai

Trijų berniukų pavyzdy
$S_1 = \{\{s_1, s_2\}, \{s_1, s_3\}, S_2 = \{s_2, s_3\}, \{s_1, s_2\}, S_3 = \{s_2, s_3\}, \{s_1, s_3\}\}$.
Iš (207)(197) Shapley dalybos

$$\phi_i[v] = (2-1)!(3-2)!/3! + (2-1)!(3-2)!/3! = 1/3. \qquad (208)$$

UAB pavyzdy
$S_1 = \{1, 2, 3\},\ S_2 = \{\{1, 2, 3\}, \{2, 4\}, \{1, 2, 4\}\}$,
$S_3 = \{\{1, 2, 3\}, \{3, 4\}, \{1, 3, 4\}\}$,
$S_4 = \{\{2, 4\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$.
Ǐ(207)(200)

$$\begin{aligned}
\phi_1[v] &= (2-1)!(4-1)!/4! = 1/12, \\
\phi_2[v] &= 2(3-1)!(4-3)!/4! + (2-1)!(4-2)!/4! = 3/12, \\
\phi_3[v] &= 2(3-1)!(4-3)!/4! + (2-1)!(4-2)!/4! = 3/12, \\
\phi_4[v] &= 3(3-1)!(4-3)!/4! + 2(2-1)!(4-2)!/4! = 5/12.
\end{aligned}$$

Dalybų stabilizacija

Stabilios tik branduolio dalybos,
nes kiekvienam aišku, kad jų neapsimoka keisti.
Shapley dalybos bus stabilios
tik su papildoma sąlyga,
kad visi lošėjai numato partnerių reakciją
ir jos pasėkas.
Butent tai įvertina Shapley aksiomos.
Dalybas galima stabilizuoti baudomis
arba premijomis.

Stabilizacijos pavyzdys 1

Trijų berniukų lošime įvesim "vienybės " pemiją: po vienetą kievienam. Tada, išlošimų lentelėje (196) atsiras ketvirtas stulpelis skirtas koalicijai $s_4 = \{1, 2, 3\}$

$$u(i,j) = \left| \begin{array}{cccc} 1 & 1 & -2 & 1 \\ 1 & -2 & 1 & 1 \\ -2 & 1 & 1 & 1 \end{array} \right|. \qquad (209)$$

Lošimo charakteringa funkcija

$$v(s) = \begin{cases} 2, & \text{jei } |s| = 2 \\ -2, & \text{jei } |s| = 1 \\ 3, & \text{jei } |s| = 3 \ . \end{cases} \qquad (210)$$

Lošimas v ne su pastovia suma

$$v(s) + v(S \setminus s) \neq v(S) = 3, \ |s| < 3. \quad (211)$$

Čia branduolį $C(v) = (1, 1, 1)$ realizuoja koalicija $s_4 = \{1, 2, 3\}$.

Stabilizacijos pavyzdys 2

UAB lošime įveskim baudą -1
tai koalicijai, kurios nariai nesilaiko įstatymu
numatytų pelno dalybų
proporcingai akcijų skaičiui.
Lošimo charakteringa funkcija

$$v(s_j) = \begin{cases} 1, & \text{jei } G_j > 50 \text{ ir } z_i \in Z,\ i \in s_j \\ 0, & \text{jei } G_j > 50 \text{ ir } z_i \bar{\in} Z,\ i \in s_j \ . \\ -1, & \text{jei } G_j \leq 50 \text{ ir } z_i \bar{\in} Z,\ i \in s_j \ . \end{cases}$$

Čia Z proporcingos dalybos. Tada baranduolį
$C(v) = (10, 20, 30, 40)$
realizuoja koalicija $s = \{1, 2, 3, 4\}$.

AR-ABS modelis

Tai auto-regresijos modelis

$$w_t = \sum_{i=1}^{p} a_i w_{t-i} + \epsilon_{t-j} + \epsilon_t, \qquad (212)$$

kur

w_t prognozė rytoj

w_{t-1} ši diena,

ϵ_t atsitiktinė paklaida rytoj,

a_i įtakos koeficientai,

juos nustatom minimizuodami

$$f(x) = \sum_{t=1}^{T} |\epsilon_t|. \qquad (213)$$

Čia tinka tiesinis programavimas

$$\min_{a,u} \sum_{t=1}^{T} u_t \qquad (214)$$

$$u_t \geq \epsilon_t,\ t = 1, ..., T, \qquad (215)$$

$$u_t \geq -\epsilon_t\ t = 1, ..., T, \qquad (216)$$

$$a_i = a_i^1 - a_i^2,\ i = 1, ..., p. \qquad (217)$$