

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

# Teltonika gamyklos įrenginių gamybos modeliavimas

Andrej Bugajev

2021-06-28

## Tyrimo objektas

Tyrimo objektas – Teltonika įrenginių gamykla

## Problema

Šiuo metu esantis gamybos procesų valdymas yra neefektyvus – dėl didelio užsakymų kiekio šiuo metu gamykloje taikomi metodai neleidžia sudaryti tinkamos kokybės darbų tvarkaraščio.

## Tikslas

Modeliavimo tikslas – pagerinti gamybos procesų valdymą simuliacijos ir tam tikrų valdymo dalių automatizacijos pagalba.

## Uždaviniai

- 1 Sukurti pagrindinių gamybos procesų simuliaciją.
- 2 Optimizuoti darbų tvarkaraštį.

## Užsakymai ir simuliacija

$T_0$  yra modeliavimo pradžios laikas. Planą sudaro užsakymai  $U_i, i = 1, \dots, N$ ,  $U_i = (k_i, n_i, T_i)$ , kur  $U_i$  yra  $i$ -tasis užsakymas, kurį sudaro  $k_i$  – gaminio kodas,  $n_i$  – kiekis,  $T_i$  – užsakymo termino laikas.  $i$ -tojo užsakymo vykdymą apibūdina laiko intervalas  $[a_i, b_i], i = 1, \dots, N$ ,  $a_i$  – nurodo kokiu laiko momentu užsakymą reikia pradėti vykdyti,  $b_i$  – užsakymo vykdymo pabaigos laikas.

### Simuliacija

Modeliavimą (proceso simuliaciją) apibūdinysime funkcija  $S$ :

$$(b_1, b_2, \dots, b_N) = S(a_1, a_2, \dots, a_N), \quad (1)$$

t.y. ši funkcija paskaičiuoja produktų pabaigos laikus iš pradžios laikų.

# Optimizavimo uždavinys

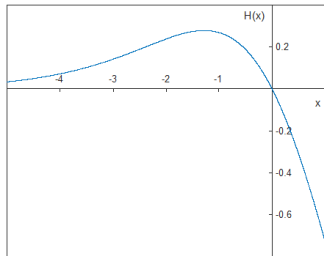
Tada uždavinys – rasti tokį  $(a_1, a_2, \dots, a_N)$  rinkinį, kad  $(b_1, b_2, \dots, b_N)$  kuo geriau tenkintų terminų grafiką  $(T_1, T_2, \dots, T_N)$ . Kadangi užsakymų vėlavimas yra nepageidaujamas reiškinys, todėl įvesime specialią vėlavimo atvaizdavimo funkciją  $H(x)$ , kurios argumento reikšmė bus skaičiuojama  $x = b_i - T_i$ . Šią funkciją reikia sudaryti atsižvelgiant į tai, kad:

- 1 užsakymą pageidautina paruošti anksčiau nei numatytąja data, pavyzdžiui, su 3 dienų atsarga;
- 2 užsakymo vėlavimas yra blogiau už pernelyg anksčiau įvykdytą užsakymą;
- 3 baudos (mažos funkcijos reikšmės) esant dideliame nukrypimui turi būti protingos, kad, pavyzdžiui, esant būtinybei modelis palaikytų vėlavimą ir adekvačiai reaguotų į tokias situacijas.
- 4 kitokios sąlygos.

## Modeliuojamos funkcijos pavyzdys

$$H(x) = -\frac{xe^{k(x-x_0)}}{1 + e^{k(x-x_0)}}, \quad (2)$$

kurios atskiras atvejis, kai  $k = 1, x_0 = 0$  pateiktas žemiau.  $H(x)$  galutinis pavidalas bus nustatytas tyrimų eigoje.



pav.:  $H(x)$  funkcija, kai  $k = 1, x_0 = 0$

# Optimizavimo uždavinys

Tada formuluokime tokį optimizavimo uždavinį

$$\max_{a_i \in [T_0, \infty), i=1, \dots, N} \frac{1}{N} \sum_{i=1}^N H(b_i - T_i), \quad (3)$$

kur  $(b_1, b_2, \dots, b_N) = S(a_1, a_2, \dots, a_N)$ .

Kaip matome, sekantis žingsnis yra apibrėžti  $S$  – gamyklos simuliacijos procedūrą su turimais apribojimais ir tam tikromis prielaidomis.

## Alternatyvi uždavinio formuluotė

Jeigu nelaikykime pernelyg anksti atliktus darbus neoptimaliais, tada galima nagrinėti, pavyzdžiui, tokią monotonišką funkciją:

$$H(x) = \begin{cases} 0, & \text{kai } x \leq x_0 \\ -\left(\frac{(x - x_0)}{x_0}\right)^4, & \text{kai } x > x_0. \end{cases} \quad (4)$$

Šioje funkcijoje tiksliai laiku  $x_0$  atliktas darbas duos reikšmę  $-1$ , o didelis laipsnis  $4$  reiškia baudos už vėlavimą netiesini didėjimą.

## Alternatyvi uždavinio formuluotė

Remsimės šiomis prielaidomis:

- 1 Darbo pradžios dirbtinis uždelsimas, kai kartu su esamais nagrinėjamo etapo darbais darbas gali būti vykdomas be prastovų visuose likusiuose etapuose, ženkliai nepagerins rezultato.
- 2 Kodų gamybos tvarkaraštį vienareikšmiškai nulemia eiliškumas, kuriuo pradedami darbai.
- 3 Visus užsakymus su sutampančiais kodais grupuojame į vieną darbą.

Tada uždavinio formuluotę (3) galime pakeisti į optimalios darbų tvarkos sudarymą, kurioje tokie patys kodai eis iš eiles vienas po kito, o sprendinį apibrėžia kodų tvarka:

$$\bar{Z} = (k_{i_1}, k_{i_2}, \dots, k_{i_{\bar{N}}}), \quad (b_1, b_2, \dots, b_N) = \bar{S}(\bar{Z}, U). \quad (5)$$

čia  $\bar{N}$  – skirtingų kodų skaičius.



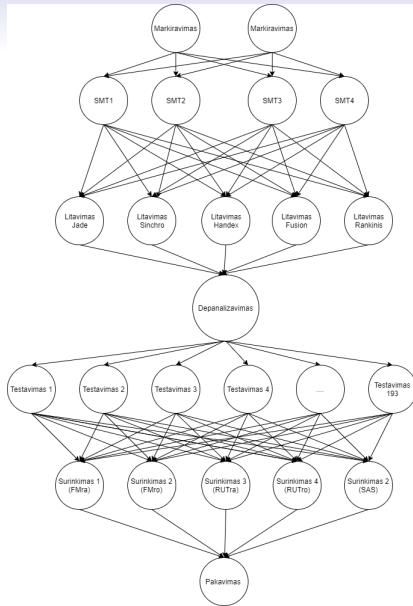
## Gamybos proceso formalizavimas

- Kiekvienam gaminiui pagaminti reikalinga darbų seka,
- kiekvieną iš šių darbų gali atlikti tik tam tikra vykdytojų (gamyklos įrenginių) aibė,
- tas pats vykdytojas moka atlikti nedaugiau nei vieną iš šios sekos darbų.

Gaminiai apdorojami porcijomis, modeliuojant darbu laikomas porcijos apdorojimas, jo atlikimo laikas skaičiuojamas pagal formulę

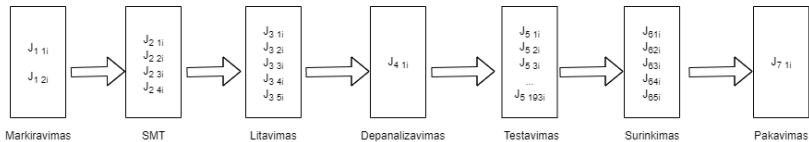
$$\alpha + \beta n, \quad (6)$$

čia  $n$  – gaminių porcijoje kiekis,  $\beta$  – duotojo darbo operacijų sekoje lėčiausios operacijos laikas,  $\alpha$  – laikas, gautas atėmus  $\beta$  iš vieno gaminio pagaminimo laiko.  $\alpha$  ir  $\beta$  priklauso nuo kodo ir įrenginio (gali turėti skirtingus našumus). Darbu vadinsime porciją arba jos dalį, kuria šiuo metu apdoroja vykdytojas arba kuri laukia eilėje, t.y. darbas apibūdinamas gaminių skaičiumi  $n$  ir kodu  $k$ :  $J = (k, n)$ .



pav.: Gamybos proceso grafas, kai kodas turi visas įmanomas briaunas

Gamybos procesą galime apibūdinti etapais  $E_m$ ,  $m = 1, \dots, 7$ :  
 markiravimas, SMT, litavimas, depanelizavimas, testavimas,  
 surinkimas ir pakavimas. Kiekvienas etapas susidaro iš  
 darbininkų  $d_{ms}$ ,  $s = 1, \dots, N$ , kurie gali vykdyti šio etapo  
 darbus.



pav.: Gamybos proceso etapai

## Etapas

Etapą apibėžia:

- Jo numeris bendroje etapų sekoje
- Darbininkai (vykdytojai)
- Šiuo (tam tikru modeliavimo) metu esančių eilėje (laukiančių) darbų aibė.

## Darbininkas

Darbininką apibėžia:

- darbininko atliekamų darbų sąrašas, su duomenimis: kodas,  $\alpha$ ,  $\beta$ , changeover laikas<sup>1</sup>, porcijos dydis (rackas).
- jo būseną: laisvas, užimtas arba vykdo darbą bet gali pradėti naują (atlaisvintas buferis),
- vykdomų darbų eilė,

---

<sup>1</sup>Papildomas laikas, reikalingas darbininko perkonfigūravimui, kai prieš tai vykdytas darbas buvo kitoks

Taigi gamykla yra sistema:

- pagrindiniai šios sistemos komponentai – etapai, jie susieti tarpusavyje tvarka – tam tikrame etape įvykdyti darbai patenka į sekantį etapą,
- etapus sudaro sandėliai su laukiančiais darbais ir darbininkai,
- darbininkai apibūdinami informacija apie jų galimybes ir vykdomus darbus.

Sistemos veikimą aprašysime ECA (Event, Condition, Action) taisyklėmis:

### ① Taisyklė 1

- Įvykis: praeito etapo darbininkas užbaigia darbą ir darbas atsiranda etape su darbininkais  $W$ .
- Sąlyga: tarp darbininkų  $W$  yra keli laisvi galintys atlikti naują darbą.
- Veiksmas: darbininkai iš eilės pilnai (arba dalinai, jeigu trūksta) užpildo savo darbų buferius, kol tai leidžiama atsižvelgiant į ateities etapų pralaidumą.

### ② Taisyklė 2

- Įvykis: atsilaisvina darbininko buferis.
- Sąlyga: etapo sandėlyje yra laisvų darbų, naujo darbo kodas yra toks pats ir darbininkas yra laisvas arba užbaigiantis darbą.
- Veiksmas: registruojami du įvykiai
  - ① darbininko buferio atsilaisvinimo įvykis laiko momentu
$$T_1 = n * \beta$$
  - ② darbininko darbo porcijos baigimo įvykis laiko momentu
$$T_2 = \alpha + n * \beta$$

### 3 Taisyklė 3

- Įvykis: atsilaisvina darbininko buferis.
- Sąlyga: etapo sandėlyje yra laisvų darbų, darbininkas yra laisvas ir naujo darbo kodas skiriasi nuo prieš tai buvusio.
- Veiksmas: registruojami du įvykiai su laikais

$$T_1 = \text{changeover} + n * \beta \text{ ir } T_2 = \text{changeover} + \alpha + n * \beta$$

Grindžiamo įvykiais proceso modeliavimas paprastai realizuojamas diskrečiųjų įvykių simuliacijos metodu, kuris turi dvi bendras realizacijas:

- 1 Laiko inkrementų simuliacija – laikas diskretizuojamas, algoritmas iteruoja per laiko žingsnius tikrindamas sąlygų išpildymą.
- 2 Sekančio įvykio simuliacija (angl. next-event simulation) – remiasi prielaida, kad sistemos būseną gali pasikeisti tik įvykius įvykiams.

## Sekančio įvykio simuliacija

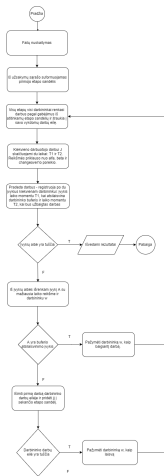
- Jokių skaičiavimų kitokiais nei įvykių momentais nevyksta.
- Paprastai tai yra žymiai greitesnis modeliavimo būdas.
- Modeliavimas grindžiamas įvykių saugojimu duomenų struktūroje, leidžiančioje rūšiuoti joje saugojamus įvykius pagal laiką, pavyzdžiui dvejetainėje piramidėje.

Šiuo atveju įvykis aprašomas šiais parametrais:

- 1 laikas
- 2 tipas:
  - 1 READY – buferio atlaisvinimo įvykis (gali pradėti darbą)
  - 2 FINISHED (darbas perduodamas sekančiam etapui)
- 3 darbininkas  $w$ , su kuriuo yra susietas šis įvykis



# Sprendimo algoritmas



pav.: Programos veikimo schema

## Kai kurios praleistos detalės

- 1 Jeigu užsakymų sąrašė yra keli sutampantys kodai – juos grupuojame tardami, kad gaminsime vieną po kito. Daroma prielaida, kad modeliuojamas sąlyginai nedidelis laikotarpis (apie savaitę).
- 2 Kai tik atsiranda poreikis daryti changeover ir darbininkas tam tikrą laiką nieko nedarė: jeigu prastova yra trumpesnė už changeover laiką, uždelsti darbo pradžia changeover ir prastovos laikų skirtumu kitu atveju changeoverio pradžia nustumti į praeitį per changeover laiką. Tam tikslui įvedamas naujas įvykio tipas – POSTPRONED (įvykio nukėlimas į ateitį).
- 3 Pralaidumo režiai – suskaičiuoti kokį pralaidumą kuriame etape yra tikslinga palaikyti, ir neleisti jo viršyti.

# Ačiū už dėmesį

Klausimai?