

Teoria

- Johdanto simulointiin
- Simuloinnin kulku -- prosessin realisaatioiden tuottaminen
- Satunnaismuuttujan arvonta annetusta jakaumasta
- Tulosten keruu ja analyysi
- Varianssinreduktioteknikoista

20/09/2004

1

Prosessin reaalisatioiden tuottaminen

- Oletetaan, että olemme mallintaneet tarkasteltavan järjestelmän stokastisena prosessina
- Seuraavana tehtävänä on prosessin realisaatioiden tuottaminen (eli järjestelmän simulointi suppeassa mielessä). Se koostuu kahdesta osasta:
 - kaikille prosessin kulkuun vaikuttaville satunnaismuuttujille on arvottava arvot (yleensä reaalityluku) satunnaisesti ko. $sm:n$ jakaumasta ($sm:ien$ väliset riippuvuudet tietyksi huomioiden)
 - näin saaduilla arvoilla konstruoidaan prosessin reaalisatio ts. sen kehittyminen ajassa
- Simuloinnissa nämä kaksi osaa eivät suinkaan tapahdu peräkkäin eri vaiheissa, vaan **limittään**:
 - välillä on siis arvottava jollekin satunnaismuuttujalle arvo, jota sitten käytetään (yhdessä aiemmin arvottujen $sm:ien$ kanssa) prosessin reaalisatioiden konstruointiin jollakin lyhyehkällä aikavälillä simuloinnin nykyhetkestä eteenpäin
 - Satunnaismuuttujien arvojen arvonta perustuu ns. **(pseudo)satunnaislukujen generointiin**
- Prosessin reaalisatioiden konstruointi tehdään yleensä **tapahtumapohjaisesti** (discrete event simulation)

20/09/2004

2

Tapahtumapohjainen simulointi (1)

- Mitkä ovat järjestelmän tilan ja havaintojen kannalta kiinnostavat suureet?
- Mitkä ovat olennaiset **tapahtumat**?
- Mitkä ovat tapahtumien laukaisijat ja niiden ajoitukset?

20/09/2004

3

Tapahtumapohjainen simulointi (2)

- Idea: simulointi etenee **tapahtumasta tapahtumaan**
 - jos jollakin aikavälillä ei tapahdu mitään, voimme hypätä ko. aikavälin yli
- Tapahtuma vastaa (yleensä) aina systeemin tilan muuttumista
 - esim. yksinkertaisessa liikenneteoreettisessa mallissa mahdollisia tapahtumia ovat ainakin asiakkaiden saapumiset ja poistumiset systeemistä
 - prosessin reaalisation generoinnin lopetus on kuitenkin oma tapahtumansa
 - samoin tietojen keruu voi aiheuttaa joitakin "ylimääräisiä" tapahtumia
- Tapahtuma karakterisoidaan kahdella parametrilla
 - tapahtumahetki (so. milloin tapahtuma käsitellään) ja
 - tapahtuman tyyppi (so. miten tapahtuma käsitellään)
- Tapahtumat organisoidaan yleensä tapahtumahetken mukaan järjestetyksi **tapahtumalistaksi** (event list), jonka kärjessä on seuraavaksi sattuva tapahtuma (siis pienin tapahtumahetki)
 - Listaa käydään läpi tapahtuma tapahtumalta (generoiden samalla uusia tapahtumia listan loppupäähän). Kun tapahtuma on käsitelty, se poistetaan listalta.
 - Simulointikello kertoo, mikä on käsiteltävänä olevan tapahtuman hetki
 - Se siis etenee hyppäyksittäin

20/09/2004

4

Tapahtumapohjainen simulointi (3)

- 1. Initialisointi
 - aseta simulointikello nolllaksi
 - aseta systeemin tila valittuun alkuarvoonsa
 - generoi kunkin tapahtumatyyppin seuraava tapahtuma (mikäli mahdollista) ja liitä näin saadut tapahtumat tapahtumalistaan
- 2. Simulointikellon siirto
 - aseta simulointiajaksi (tapahtumalistan kärjessä olevan) seuraavan tapahtuman tapahtumahetki
- 3. Tapahtuman käsittely
 - käsittele tapahtuma (mahdollisesti generoiden samalla uusia tapahtumia ja liittäen ne tapahtumalistaan tapahtumahetkensä mukaiseen järjestykseen) sekä päivitä systeemin tila
 - poista käsitelty tapahtuma tapahtumalistalta
- 4. Lopetusehto
 - jos lopetusehto on voimassa, lopeta prosessin reaalisaaion generointi; muutoin palaa kohtaan 2

20/09/2004

5

Esimerkki: M/M/1- FIFO jonon simulointi

- Kiinnostavia suureita:
 - odottavien (jonottavien) asiakkaiden lukumäärä
 - asiakkaan odotusaika
 - järjestelmässä olevien asiakkaiden lukumäärä
 - asiakkaan järjestelmässä viettämä aika

20/09/2004

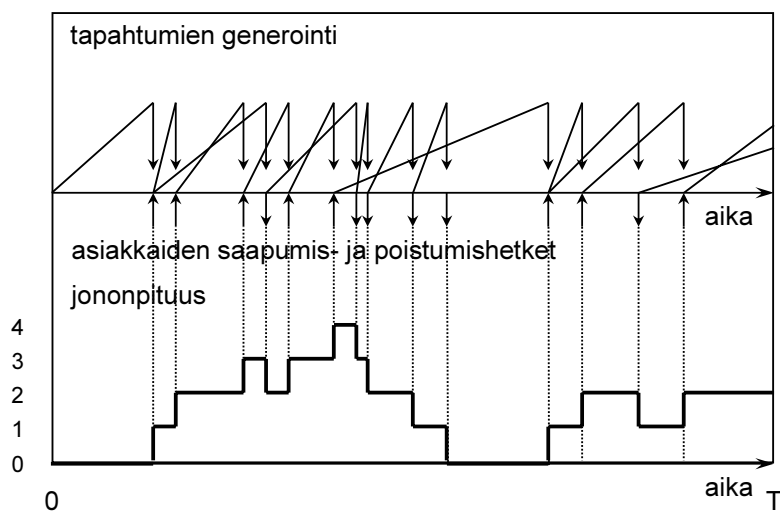
6

Esimerkki: M/M/1- FIFO jonon simulointi

- Simuloidaan M/M/1-jonon jononpituuden kehitystä ajassa hetkestä 0 hetkeen T olettaen, että systeemi on tyhjä hetkellä 0.
- Systeemin tilaa hetkellä t kuvaa siis jononpituus X_t .
- Tapahtumia ovat asiakk. saapumiset ja poistumiset sekä simuloinnin lopetus
- Initialisointi:
 - asetetaan $X_0 = 0$
 - arvotaan ensimmäisen asiakkaan saapumishetki $\text{Exp}(\lambda)$ -jakaumasta
- Tapahtuman käsittely uuden asiakkaan saapuessa (hetkellä t)
 - systeemin tilaa eli jononpituutta kasvatetaan yhdellä: $X_t = X_t + 1$
 - jos systeemi oli tyhjä asiakkaan saapuessa, generoidaan ko. asiakkaan poistumishetki $t + S$, missä S on ko. asiakkaan palveluaika (arvottu $\text{Exp}(\mu)$ -jakaumasta)
 - generoidaan seuraavan asiakkaan saapumishetki $t + I$, missä I on saapumisten väliaika (arvottu $\text{Exp}(\lambda)$ -jakaumasta)
- Tapahtuman käsittely asiakkaan poistuessa (hetkellä t)
 - systeemin tilaa eli jononpituutta vähennetään yhdellä: $X_t = X_t - 1$
 - jos systeemiin jäi asiakkaita, generoidaan seuraavaksi palveltavan asiakkaan poistumishetki $t + S$, missä S on ko. asiak. palveluaika (arvottu $\text{Exp}(\mu)$ -jakaumasta)
- Lopetusehto: $t > T$

20/09/2004

7

Jononpituuden kehitys M/M/1-FIFO jonossa

20/09/2004

8

Simulointiohjelman komponentit

1. Tapahtumien skeduloija (tapahtumalistan ylläpitäjä)

- ylläpitää linkitettyä listaa tulevista tapahtumista
- skeduloija voi muokata tapahtumalistaa, esim.
 - skeduloi tapahtuman e tapahtumaan hetkellä t
 - kumoaa aikaisemmin skeduloidun tapahtuman e (poistaa listasta)
 - pidättää tapahtumaa e aikaintervallin Δt
 - pidättää tapahtumaa e toistaiseksi (kunnes toinen tapahtuma skeduloi sen)
 - skeduloi toistaiseksi pidätetyn tapahtuman
- on simulointiohjelman ydin, ohjelman useimmin suoritettu komponentti
- suoritetaan aina ennen tapahtumaa
- skeduloijaa voidaan kutsua useita kertoja tapahtumankäsittelyn yhteydessä skeduloimaan uusia tapahtumia

20/09/2004

9

Simulointiohjelman komponentit

2. Simulointikello ja ajansiirtomekanismi

- jokaisessa simulaattorissa on globaali muuttuja, joka edustaa aikaa simuloidussa järjestelmässä
- ajan siirtämiseen voidaan käyttää kahta eri tapaa
 - a) siirto vakioinkrementein
 - aika jaetaan pieniin askeliin
 - siirrytään eteenpäin askel kerrallaan
 - käsitellään kerrallaan kaikki tapahtumat, jotka ovat sattuneet askeleen sisällä
 - b) tapahtumapohjainen siirto
 - aika siirretään osoittamaan lähinnä seuraavaan tapahtumaan
- menettely b) on yleisempi ja dynaamisempi
- vältetään ongelmat, jotka aiheutuvat useiden tapahtumien osumisesta samaan aikaväliin
- toisaalta ei tarvitse ”kelata tyhjää” kuten joudutaan tekemään menetelmässä a) jos aikaväli on valittu hyvin lyhyeksi samanaikaisten tapahtumien välttämiseksi

20/09/2004

10

Simulointiohjelman komponentit

3. Järjestelmän tilamuuttujat

- globaalit muuttujat, jotka kuvaavat järjestelmän tilaa
- esim. asiakkaiden lukumäärä jonossa
 - eri kuin lokaali muuttuja, esim. asiakkaan vaatima palvelun kesto, joka voidaan tallettaa asiakkaan yhteyteen

4. Tapahtumankäsittelijät

- kunkin tapahtuman käsittely suoritetaan omalla rutiinillaan
- rutiinit päivittävät tilamuuttujien arvoja ja skeduloivat uusia tapahtumia (kutsuvat tapahtumalistan ylläpitäjää)
- esim. asiakkaan saapumisen käsittelijä tai poistumisen käsittelijä

5. Syöttörutiinit

- simuloinnissa tarvittavien lähtöarvojen syöttörutiinit
- kysellään käyttäjältä parametrien arvot tai
- vaihteluvälit ja askeleet kartoitettaessa jonkin parametrin vaikutusta

20/09/2004

11

Simulointiohjelman komponentit

6. Raportin generaattori

- kerää simuloinnin kuluessa tilastoa halutuista muuttujista
- käsittelee tulokset tilastollisesti
- tulostaa halutussa muodossa

7. Initialisointirutiinit

- asettaa tilamuuttujien alkuarvot ja satunnaislukustreamien alkuarvot
 - oma rutiini, jolla järjestelmä alustetaan annettuihin parametrien arvoihin
 - oma rutiini, jolla ohjelma alustetaan samoilla parametrien arvoilla tehtävien toistojen alussa (laskurien nollaukset jne)

8. Trace-rutiini

- tulostavat välimuuttujien arvoja simuloinnin varrelta
- helpottaa ohjelman debuggausta
- on/off-kytkin: voidaan kytkeä pois päältä tuotantoajoja tehtäessä

20/09/2004

12

Simulointiohjelman komponentit

9. Dynaaminen muistinhallinta

- simuloinnin kuluessa systeemi ”elää”
- uusia olioita tulee ja poistuu
- simulointikielissä ja nykyisissä yleiskielissä ohjelmoijan ei tarvitse tähän juurikaan puuttua

10. Pääohjelma

- sitoo ohjelman eri osat toisiinsa
- kutsuu syöttörutiineja ja intialisointirutiineja
- aloittaa simulointisilmukan
- lopulta kutsuu raportingeneraattoria