PIENI SIMULOINTIKIRJA

Simuloinnin alkeita taulukkolaskennan avulla

Matti Lehtinen

Maanpuolustuskorkeakoulu
Tekniikan laitos
Julkaisusarja 5
No 3

Maanpuolustuskorkeakoulu, Tekniikan laitos
ISBN 951-25-1555-5
ISSN 1795-3294
Painopaikka: Edita Prima Oy
Helsinki 2004

SISÄLLYS

1 JOHDANTO

Tässä vihkosen työnimi on ollut *Simulointia joka pojan menetelmin.* Tarkoitus on siis esittää muutamia yksinkertaisia sotaväen toimintaa sivuavia simulointimalleja. Niitä yhdistävä tekijä on mahdollisuus toteuttaa simuloinnit työkalulla, joka on helposti kaikkien saatavilla, nimittäin taulukkolaskentaohjelmistolla, ja ilman erityisiä ohjelmointitaitoja. Taulukkolaskentasimulointi antaa yksinkertaisuudestaan ja käsityöläismäisyydestään huolimatta tuntumaa joihinkin simuloitavien ilmiöiden olennaisiin piirteisiin. Simulointimallien rakentelu on myös harjoitusta itse taulukkolaskentaan, yllättävän monikäyttöiseen laskemisen yleistyökaluun. Tietokonehan on *kompuutteri*, laskija. Taulukkolaskenta on mitä suurimmassa määrin läpinäkyvää, koska välitulokset ovat kaikki taulukossa. Tämä tekee ohjelman oikeellisuuden verifioinnista yleensä helppoa.

Esitetyt mallit on tarkoitettu stimuloiviksi, ei suoraan simuloitaviksi (latinan *simulare* = matkia). Esitetyt esimerkit ovat melko satunnaisia. Tavoite on, että lukija huomaa taulukkolaskennan mahdollisuudet ja ryhtyy käyttämään niitä omiin tarkoituksiinsa. Esityksen seuraaminen onnistuu kuitenkin vain, jos taulukkolaskentaohjelma on avattuna ja lukija itse täydentää pakosta osin viitteellisesti kuvatut esimerkit. Monet asiat voi toteuttaa eri tavoin. Lukija voi vapaasti itse keksiä tässä esitetyille keinoille vaihtoehtoja, ehkä paljonkin parempia.

Taulukkolaskennan perusominaisuudet kuten ruutuosoitteiden sarakekirjain- ja rivinumerorakenne, numeerisen tiedon, laskukaavan ja tekstitiedon syötön ero sekä yksinkertaisimmat funktiot kuten summa oletetaan lukijalle tutuiksi.

Esitetyt esimerkit on rakennettu käyttämällä Lotus 1–2–3:n komentoja ja syntaksia. Tekijän käyttämän ohjelmaversion taulukossa on 256 saraketta, A, B, C, ..., Z, AA, AB, ..., IV ja $2^{16} = 65536$ riviä 1, 2, ..., 65536. Muutokset Microsoft Excel -taulukkolaskentaan ovat yksinkertaisia. Tässä esitettyjen sovelluksien kannalta Lotus-tuote tuntuu hiukan kilpailijaansa joustavammalta. Esityksen tavoitteiden mukaista tarkoituksellisuutta on ollut välttää taulukkolaskentaankin sisältyvien kehittyneempien ohjelmointielementtien kuten ns. makrojen käyttöä. Taulukkolaskentaohjelmien suomenkieliset käyttöohjeet, niin sähköiset kuin painetutkin, eivät ole erityisen laadukkaita ja niihin on syytä suhtautua hiukan varoen.

Kun simuloidaan epävarmoja asioita, liikutaan aina todennäköisyyslaskennan piirissä. Monte Carlo -simulointiin liittyy tulosten tilastollisen merkitsevyyden ja luotettavuuden arviointia. Niinpä näiden esimerkkienkin läpikäynti edellyttää todennäköisyyslaskennan ja tilastomatematiikan alkeiden hallintaa.

Toisaalta simulointi luo konkretiaa abstraktiin todennäköisyyskäsitteeseen ja auttaa sen ymmärtämistä. Alan kirjallisuutta on riittämiin saatavissa. Paremman puutteessa voi näihin alkeisiin tutustua vaikkapa kirjoittajan vihkosista [3] ja [2].

Esitykseen on sinne tänne siroteltu muutamia harjoitustehtäviä. Ne on esitetty jokseenkin yleisluontoisesti: kysymys on toisinaan pienistä tutkimustehtävistä ja toisinaan tekstissä väljästi muotoiltujen menetelmien tarkennustehtävistä ennemin kuin laskurutiinin ja opitun suoran soveltamisen äksiisistä. Tarvittavien parametrien valinta on tietoisesti jätetty lukijoille. Yksi syy tehdä simulointeja on juuri parametrien muutosten vaikutuksen arviointi. Näin ollen harjoitustehtäviin ei voida esittää valmiita oikeita ratkaisuja.

Merkittävänä johdatuksena yksinkertaiseen simulointiin on tekijälle toiminut amerikkalaisen *Sheldon M. Rossin* pieni oppikirja [4]. Useita kirjassa käsiteltyjä asioita on esitelty myös vihkosessa [1]. Tätä esitystä voi halutessaan pitää yrityksenä liittää [1]:een työkirja.

2 TAULUKKOLASKENTA JA SIMULOINTI

Tässä luvussa esitellään lyhyesti niitä taulukkolaskentatyökalun ominaisuuksia, joita myöhemmissä esimerkeissä käytetään hyödyksi.

2.1 Kaavan kopiointi ja liittäminen

Ohjelmoinnissa olennainen toistorakenne asustaa taulukkolaskennan kaavan kopiontitoiminnossa. Mikä tahansa taulukon solussa oleva kaava, jonka syötteissä on viittauksia taulukon muihin soluihin, voidaan kopioida taulukon toiseen paikkaan kaavaksi, jonka syötteiden suhteellinen sijainti on sama. Kopiointi tehdään joko muokkausvalikon kopioi- ja liitä -toiminnoilla tai suoraan näppäimistön Ctrl C- ja Ctrl V -toiminnoilla. Jos useat kaavat tarvitsevat tiettyyn kiinteään soluun tallennetun parametrin, on viittauksissa tämän solun osoitteessa sekä sarakkeen kirjaimen että rivin numeron eteen kirjoitettava \$-merkki. Jos solussa oleva kaava sisältää viittauksia saman rivin tietyn sarakkeen ruutuihin ja saman sarakkeen tietyn rivin ruutuihin, laitetaan \$-merkki vain rivinumeron eteen tai vain sarakekirjaimen eteen.

2.2 Ehtolause

Taulukkolaskennan ehtolause on muotoa @IF(*ehto; kaava, joka on voimassa jos ehto toteutuu; kaava, joka on voimassa, jos ehto ei toteudu*). Useampien vaihtoehtojen kesken tapahtuva valinta voidaan rakentaa useammasta sisäkkäisestä @IF-lauseesta. Esimerkiksi tilanne, jossa lasketaan $kaava_1$, jos parametrilla x on arvo 1, $kaava_2$, jos x:n arvo on 2 ja $kaava_3$, jos x ei ole 1 eikä 2, voidaan kirjoittaa @IF($x = 1$; $kaava_1$; @IF($x = 2$; $kaava_2$; $kaava_3$))

Ehto on yleensä jokin yhtälö tai epäyhtälö. Relaation ”eri suuri kuin” merkki on <>, ”pienempi tai yhtä suuri kuin” <= ja ”suurempi tai yhtä suuri kuin” >=. Useammasta yksinkertaisesta yhtälöstä tai epäyhtälöstä voi koota yhden ehdon käyttämällä loogisia operaattoreita #JA# ja #TAI#. (Excel-ohjelmassa loogisten operaattorien tilalla ovat hiukan eri tavalla järjestyvät loogiset funktiot JA ja TAI.)

2.3 Satunnaisluvut

Simuloinnin yksi keskeisiä osioita on sattumanvaraisten ilmiöiden matkiminen. Keinotekoinen sattumanvaraisuus synnytetään *satunnaislukujen* avulla. Satunnaisluvun ymmärtäminen vaatii satunnaismuuttujan jakauman käsitteen hallintaa. Todennäköisyyslaskennassa usein noudatettavan käytännön mukaisesti satunnaislukuja niin kuin satunnaismuuttujia yleensäkin merkitään kapiteelikirjaimin X, Y jne.

Mikään tietokoneen tai laskulaitteen ohjelmoidusti tuottama ei ole aidosti satunnaista. Satunnaislukujen generointi tietokoneessa perustuu aina johonkin lukujonoon, jossa lukua seuraavan luvun tuottaa jokin sääntö, jonka tuotos ei näytä olevan arvattavissa. Säännöt perustuvat usein sopivasti valittujen jakolaskutoimitusten jakojäännöksiin. Tässä esitettyjen esimerkkien kannalta ohjelman käyttämällä satunnaislukugeneraattorin lajilla ei liene merkitystä. Kirjallisuus tuntee kyllä tapauksia, joissa satunnaislukugeneroinnin tapa ja satunnaislukujen käyttökohde ovat ikävästi vaikuttaneet toisiinsa niin, että tulokset ovat kaukana aidosti satunnaisista.

2.3.1 Jatkuvat jakaumat

Taulukkolaskennan funktiokomento @RAND tuottaa satunnaislukuja, jotka ovat tasan jakautuneita välille [0, 1]. Jos on tarpeen tuottaa satunnaislukuja, jotka ovat tasan jakautuneita välille $[a, b]$, niin voidaan käyttää laskukaavaa $a + (b - a)*$@RAND.

Satunnaisluvut, joiden jakauma on jokin muu kuin tasainen jakauma, ovat hiukan ongelmallisempia. Jos jakauman kertymäfunktio on F ja X on välille [0, 1] tasan jakautunut satunnaismuuttuja, niin $F^{-1}(X)$ on satunnaismuuttuja, jonka kertymä on F. Jos siis jakauman kertymän käänteisfunktio hallitaan, voidaan sen ja @RAND-funktion avulla tuottaa jakaumaa noudattavia satunnaislukuja. "Satunnaisin aikavälein" tapahtuvien ilmiöiden tapahtumien väliajat noudattavat eksponenttijakaumia. Eksponenttijakauman kertymäfunktio on

$$F(x) = 1 - e^{-x/a},$$

missä a on tapahtumien väliajan odotusarvo. Tästä on helppo ratkaista $F^{-1}(y) = -a \ln(1 - y)$. Eksponentiaalisia satunnaislukuja, joiden odotusarvo on a, voi siis tuottaa komennolla $-a*$@LN(1−@RAND) tai yhtä hyvin $-a*$@LN(@RAND).

Taulukkolaskennassa on valmistoimintana normaalijakauman kertymä ja sen käänteisfunktio; Lotus 1–2–3:n komento @NORMAL(@RAND; μ; σ; 1) tuottaa normaalijakautuneita satunnaislukuja, joiden odotusarvo on μ ja keskihajonta σ.

Jos satunnaislukuja tarvitaan kerralla paljon, niin kannattaa käyttää ns. Box–Müllerin muunnokseen perustuvia kaavoja. Box–Müllerin kaavojen pohjana on havainto, että tason normaalistetussa kaksiulotteisessa symmetrisessä normaalijakaumassa eli ympyräjakaumassa pisteen napakoordinaattiesityksen (r, ϕ) kulmasuure ϕ on tasan jakautunut välille $[0, 2\pi)$, kun taas suureen r^2 jakauma on eksponenttijakauma, parametrina $a = 2$. Jos siis U ja V ovat välille [0, 1] tasan jakautuneita satunnaislukuja, niin kaavoilla

$$X = \sqrt{-2 \ln U} \cos(2\pi V) \tag{1}$$

ja

$$Y = \sqrt{-2 \ln U} \sin(2\pi V) \tag{2}$$

määritelty piste (X, Y) on tason symmetrisen normaalijakauman mukainen satunnaispiste ($\sigma = 1$), eli määritellyt X ja Y ovat riippumattomia normaalistettua normaalijakaumaa noudattavia satunnaislukuja. Silloin tietysti $\mu+\sigma X$ ja $\nu+\tau Y$ ovat satunnaislukuja, joiden jakaumat ovat (μ, σ)- ja (ν, τ)-normaalijakaumat. Riippumattomuus merkitsee, että kaavojen avulla voi arpoa tason umpimähkäisiä kaksiulotteista normaalijakaumaa (jonka hajontakuvion pääakselit ovat koordinaattiakselien suuntaisia) noudattavia pisteitä. Lotus 1–2–3:n komentoja ovat

$$\text{@SQRT}(-2*\text{@LN}(\text{@RAND}))*\text{@COS}(2*\text{@PI}*\text{@RAND})),$$
$$\text{@SQRT}(-2*\text{@LN}(\text{@RAND}))*\text{@SIN}(2*\text{@PI}*\text{@RAND})).$$

Koska molemmissa lausekkeissa selvitään samalla satunnaislukuparilla, on järkevää generoida kaksi satunnaislukua vierekkäisiin ruutuihin ja käyttää molemmissa komennoissa viittauksia näihin, niin että molemmissa logaritmitermeissä on sama viite ja kosini- ja sinitermeissä sama viite.

2.3.2 Diskreetit jakaumat

Diskreetin tasajakauman omaavien satunnaislukujen konstruointi taulukkolaskennassa on helppoa. Jos satunnaisluvun arvot ovat k, $k+1$, $\ldots n$, ja kaikkien tulisi olla yhtä todennäköisiä, niin pyöristysfunktiota @INT käyttäen saadaan jakauma asettamalla komennoksi

$$k+\ \text{@INT}((n-k+1)*\text{@RAND}).$$

(Lotuksen pyöristysfunktio @INT toimii halutulla tavalla vain positiivisten lukujen parissa. Jos satunnaislukujen joukkoon pitäisi saada negatiivisia, olisi @INT korvattava funktiolla @ROUNDDOWN.) Tavallista arpanoppaa voi siis simuloida komennolla 1+@INT(6∗@RAND).

Muiden kuin tasaisia jakaumia noudattavien diskreettien satunnaislukujen tuottaminen taulukkolaskennassa on hiukan kömpelöä. Tarvitaan tieto eri arvojen todennäköisyyksistä ja mekanismi, joka selvittää, mihin todennäköisyyshaarukkaan tasan jakautunut satunnaisluku, siis funktion @RAND tuottama luku, kuuluu.

Luultavasti tavallisin epätriviaali diskreetti jakauma on binomijakauma. Yksi tapa tehdä (n, p)-binomijakautuneita satunnaislukuja ("onnistumisten lukumäärä n:ssä riippumattomassa yrityksessä, kun yksittäisen yrityksen onnistumistodennäköisyys p") on käyttää hyväksi sen kertymäfunktiota eli kumuloituvaa binomitodennäköisyyttä. Voidaan esimerkiksi sijoittaa satunnaisluku @RAND tiettyyn ruutuun x, luvut 0:sta n:ään yhteen sarakkeeseen, ruutuihin x_k, niiden viereen, ruutuihin y_k, kumuloituva binomitodennäköisyys @BINOMIAL(n; k; p; 1) ja vielä viereen kaksinkertainen ehto @IF($x < y_k$;@IF($y_{k-1} < x$); x_k; 0). Se tuottaa sarakkeeseen nollia ja yhden nollasta eroavan luvun, joka on se onnistumisten määrä, jota @RAND:in edustama todennäköisyys vastaa. Sarakkeen summa on näin ollen haluttua binomijakaumaa noudattava satunnaisluku.

Enemmän simuloinnin hengen mukainen (n, p)-binomijakautuneiden satunnaislukujen tuottamistapa on tietysti binomijakauman itsensä simulointi: arvotaan n:ään ruutuun luku 1 todennäköisyydellä p ja 0 todennäköisyydellä

$1-p$, siis esimerkiksi komennolla @IF(@RAND< p; 1; 0), ja lasketaan ruuduissa olevien lukujen summa.

Harjoituksia

1. Poisson-jakautuneen satunnaismuuttujan X määrittelee todennäköisyys

$$P(X=k)=\frac{a^k}{k!}e^{-a}.$$

Tässä a on jakaumaa karakterisoiva parametri, itse asiassa X:n odotusarvo. Mukaile edellä esitettyä binomijakautuneiden satunnaislukujen tuottamismenetelmää niin, että se tuottaa Poisson-jakautuneita lukuja.

2.4 Laskuri

Monte Carlo -simuloinnin tarkoituksena on saada tilastollista tietoa simuloidun ilmiön lopputuloksen jakaumasta. Kukin simulointikierros tuottaa tuloksen, joka on muotoa "onnistui"/"epäonnistui" tai "päädyttiin tilaan T_j". Jos kunkin simulointikierroksen tulos voidaan toteuttaa yhdellä rivillä, kuten esimerkiksi osuiko tietyn jakauman omaava laukaus tiettyyn maalialueeseen, on yksinkertaista muodostaa lopputulokselle oma sarakkeensa, johon generoidaan 1 onnistumisen ja 0 epäonnistumisen tapauksessa. Sarakkeen summa kertoo silloin suoraan onnistumisien määrän niin monella kierroksella kuin tilanteessa on rivejä. Vastaavasti eri tiloihin johtavat simuloinnit voidaan hoitaa perustamalla kutakin lopputilaa varten oma sarakkeensa ja generoimalla sopivalla @IF-komennolla 1 siihen sarakkeeseen, jonka mukaiseen tilaan simulointi päättyi. Tilajakauma voidaan tästä määrittää.

Jos simulointi toteutetaan taulukkolaskennassa monilla riveillä, saattaa olla tarpeen synnyttää eri kierrokset uudelleenlaskentakomennolla, joka Lotuksen ja Excelin tapauksessa on tietokoneen F9-toimintonäppäin. Ohjelman asetuksista on tällöin syytä kytkeä automaattinen uudelleenlaskutoimitus pois. Excel-asetuksiin tulee iteraatiokierrosten määräksi asettaa 1. Tietokone toistaa laskua niin monta kertaa, kuin käyttäjä painaa F9-näpppäintä tai niin kauan, kuin käyttäjä pitää F9-näppäintä painettuna.

Simulointikierroksien lukumäärän ja simulontitulosten laskurit voi nyt rakentaa kaavoilla, jotka viittaavat kaavan sisältämään soluun. Yksinkertaisimmillaan solussa x oleva kaava 1+x lisää solun sisältöä yhdellä joka laskukierroksella. Tällä kaavalla voi tuottaa simulaatiokierrosten laskurin, jota seuraamalla voi ratkaista sen, kuinka pitkään pitää F9-näppäintä painettuna. Jos soluun y tuotetaan ehtolauseella 1 aina silloin, kun simulointitulos on "onnistunut" tai "tila T_j" ja nolla muulloin, niin solussa x oleva kaava y+x lisää x:ssä olevaa arvoa yhdellä, kun tulos on "onnistunut" tai "tila T_j". (Toki ehdon voi upottaa suoraan itseviittaussoluunkin.) Näin voidaan rakentaa simulointitulosten tilastoa. Laskurin nollaus onnistuu Lotus 1–2–3:ssa yksinkertaisesti tempulla, jossa laskurisolut kopioidaan (Ctrl C) ja liitetään (Ctrl V) itsensä päälle. Tämä keino ei näytä toimivan Excel-taulukoissa.

Samanlaista itseviittaustekniikkaa voidaan käyttää muidenkin funktioiden yhteydessä. Siitä seuraava esimerkki.

2.4.1 π:n määritys simuloimalla

Ympyrän ala on säteen neliö kerrottuna vakiolla, jolle tavallisesti käytetään merkintää π. Yksinkertainen tapa määrittää vakion π likiarvo on tarkastella ympyrää Y, jonka yhtälö on $x^2 + y^2 = 1$. Tämän ympyrän ala on π. Neliön, jonka sivut ovat suorilla $y = \pm 1$ ja $x = \pm 1$, ala on 4. Tähän neliöön sijoitetuista n:stä umpimähkäisestä pisteestä noin $\frac{\pi}{4}n$ sijaitsee ympyrän sisällä. Umpimähkäisen neliön pisteen (x, y) synnyttää kaksi välillä $[-1, 1]$ olevaa satunnaislukua. Sijoitetaan siis esimerkiksi ruutuihin A1 ja B1 komento $-1 + 2*$@RAND. Piste (x, y) on ympyrän $x^2 + y^2 = 1$ sisällä, jos $x^2 + y^2 < 1$. Ruutuun C1 sijoitettu ehtofunktio @IF(A1ˆ2 + B1ˆ2 < 1;1;0) tuottaa ruutuun C1 ykkösen tai nollan sen mukaan, onko umpimähkäispiste ympyrässä Y vai ei. Jos nyt alue A1..C1 kopioidaan koko taulukon sarakkeisiin A, B ja C, saadaan 65536 satunnaista neliön pistettä ja jokaisesta tieto siitä, onko piste ympyrässä Y vai ei. Pisteistä ympyrän sisään kuuluvien lukumäärän antaa sarakkeen C lukujen summa S ja luvun π likiarvon luku $4S/65536$.

Kokeilu näyttää, että tuloksissa on hajontaa. Eräät peräkkäiset kuusi arvontaa esimerkiksi tuottivat luvut 3,1477, 3,1343, 3,1447, 3,1424, 3, 1409 ja 3,1334. Arvontoja voidaan nopeasti tehdä huomattavasti enemmän käyttämällä kaavan itseviittaustekniikkaa. Sijoitetaan esimerkiksi ruutuun D1 kaava @SUM(C1..C65536) + D1, ruutuun E1 kaava E1 + 65536 ja ruutuun F1 kaava 4$*$ D1/E1. Taulukkoon voi vielä sijoittaa kierroslaskurin esimerkiksi kirjoittamalla G1-ruutuun kaavan 1+G1. Kun pitää F9-näppäintä painettuna niin kauan, että laskuri näyttää 100:aa (joka tämän kirjoittajan käyttämässä, edellä kuvatussa laitteistossa kestää noin 20 sekuntia), saa jo 6553600:n arvontaan perustuvan π:in likiarvon.

Harjoituksia

2. Määritä simuloimalla ellipsin ala, kun sen puoliakselien pituudet tiedetään. Neuvo: jos ellipsin keskipiste on origo, isoakseli $2a$ ja pikkuakseli $2b$, niin ellipsin yhtälö on

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Kannattanee arpoa satunnaispisteitä suorakaiteeseen, jonka mitat ovat $2a$ ja $2b$.

3. Jalkapallo- tai jääkiekko-ottelussa joukkueen tekemien maalien määrän voi olettaa noudattavan Poisson-jakaumaa, jossa parametrit heijastelevat joukkueiden tasoa. Rakenna simulaatio kahden saman- tai eritasoisen joukkueen A ja B pelituloksista (A:n voitto, tasapeli, B:n voitto.)

2.4.2 Keskiarvon ja keskihajonnan kerryttäminen

Itseviittausta voidaan käyttää myös simulointikierrosten tulosten keskiarvon ja keskihajonnan laskemiseen tilanteissa, joissa kierrokset generoidaan manuaalisesti, F9-näppäintä painaen. Jos A_n on lukujonon x_1, x_2, … n:n ensimmäisen jäsenen aritmeettinen keskiarvo ja V_n n:n ensimmäisen jäsenen varianssi, niin voidaan helposti johtaa palautuskaavat

$$A_{n+1} = \frac{n}{n+1}A_n + \frac{1}{n+1}x_{n+1} \tag{3}$$

ja

$$V_{n+1} = \frac{n}{n+1}V_n + \frac{1}{n}(A_{n+1} - x_{n+1})^2. \tag{4}$$

Näiden – ja edellä esitetyn laskuritoiminnon – avulla voidaan ylläpitää simulointikierrosten tulosten (x_n) keskiarvoa ja varianssia ja siten myös keskihajontaa ja keskiarvon keskihajontaa taulukkolaskennan kaavoin, joissa ruutuviittaus kohdistuu ruutuun itseensä. Lähtöarvot ovat tietysti $A_1 = x_1$ ja $V_1 = 0$.

Kaavojen sijoittamisessa taulukkolaskennan ruutuihin tulee ottaa huomioon ohjelman laskentajärjestys. Jos suure, jonka keskiarvon ja hajonnan kertymistä halutaan seurata, olisi ruudussa A1, ja itseensä viittaava lukumäärälaskuri ruudussa B1 olevan kaavan 1+B2 muodossa, niin keskiarvolaskuri olisi ruudussa C1 hiukan kaavasta (3) poikkeavassa muodossa

$$+(\mathrm{B1}-1)/\mathrm{B1}*\mathrm{C1} + (\mathrm{A1}/\mathrm{B1}),$$

sillä C1-ruudun arvoa n:nnellä kierroksella laskettaessa lukumäärälaskuri olisi jo ehtinyt arvoon $n + 1$. Kaavaan (4) perustuva varianssilaskuri ruudussa D1 olisi edelleen

$$@\mathrm{IF}(\mathrm{B1} = 1;\ 0;\ (\mathrm{B1} - 1)/\mathrm{B1}*\mathrm{D1} + (\mathrm{C1}-\mathrm{A1})\hat{\ }2/(\mathrm{B1} - 1)),$$

sillä muutoin jouduttaisiin nollalla jakoon.

Simuloinnin jatkamisen tai keskeyttämisen kannalta mielenkiintoinen suure on keskiarvon keskivirhe. Se on suure

$$s_n = \sqrt{\frac{V_n}{n}}.$$

Koska keskiarvo on aina likimain normaalijakautunut suure, on keskiarvon luottamusväli normaalijakaumasta johdettava kerroin kertaa s_n. Keskiarvon luotettavuusväli on 95 %:n luottamustasolla melko tasan $2s_n$ molempiin suuntiin keskiarvosta. Seuraamalla s_n:n kehitystä voi valita sopivan kohdan simulaation lopettamiselle. Luotettavuusvälin pituuden tai sen rajat voi sijoittaa taulukkoon ja lopettaa simuloinnin, kun toivottu tarkkuus on saavutettu.

Harjoituksia

4. Simuloi binomijakaumaa siten kuin edellä on esitetty ja määritä tätä kautta binomijakauman odotusarvo ja keskihajonta.

3 OSUMISEN SIMULOINTI

Tulen osuminen maaliin on yksi aseella vaikuttamisen peruselementti. Ampumatapahtumaan liittyvien monien satunnaisvaihtelulle alttiiden tekijöiden vuoksi osuminen on satunnaisilmiö. Siihen liittyvien mitattavien suureiden kuten maalista poikkeamien ajatellaan yleensä noudattavan normaalijakaumaa.

3.1 Osuminen yhteen tai useampaan pistemaaliin

Tarkastetaan seuraavaa perustilannetta. Oletetaan, että joukko pistemaaleja sijaitsee koordinaattipisteissä $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Oletetaan, että ammuksen vaikutusetäisyys on r. Laukausta pidetään osumana, jos se osuu alle etäisyyden r päähän jostakin maalipisteestä. Oletetaan vielä, että ampumasuunta laskettuna positiivisen x-akselin suunnasta positiiviseen kiertosuuntaan, siis vastapäivään, on ϕ. Oletetaan tulen iskemäkeskipisteeksi (x_0, y_0), pituushajonnaksi σ ja leveyshajonnaksi τ. Voidaan asettaa esimerkiksi seuraavat kysymykset. Mikä on yksittäisen laukauksen osumistodennäköisyys? Montako osumaa saadaan k:lla laukauksella?

Edellä on esitetty, miten taulukkolaskennassa voi tuottaa normaalistettua normaalijakaumaa noudattavia satunnaislukuja. Jos X' ja Y' ovat normaalistettua normaalijakaumaa noudattavat satunnaislukuja, niin $x_0 + \sigma X'$ ja $y_0 + \tau Y'$ kuvaavat satunnaisiskemän koordinaatteja $x'y'$-koordinaatistossa, jonka x'-akseli on ampumasuunnan mukainen ja y'-akseli ampumasuuntaan vastaan kohtisuorassa. Klassisten kiertokaavojen mukaan tällaisen satunnaisiskemän koordinaatit xy-tasossa ovat

$$X = x_0 + \sigma X' \cos\phi - \tau Y' \sin\phi \tag{5}$$

ja

$$Y = y_0 + \sigma X' \sin\phi + \tau Y' \cos\phi. \tag{6}$$

Kriteeri sille, milloin laukaus on osuma, on että maalin ja iskemän etäisyys jää pienemmäksi kuin ammuksen vaikutusetäisyys r. Osumisen tarkistamiseen riittää ehtoa

$$(X - x_k)^2 + (Y - y_k)^2 < r^2$$

testaava @IF-lause.

3.1.1 Yksittäislaukauksen osumistodennäköisyys

Yksinkertaisimmillaan osumistodennäköisyyden simulointi voidaan rakentaa niin, että taulukon rivi vastaa yhtä laukausta. Iskemäkeskipiste (x_0, y_0) ja ammunnan perustiedot σ, τ, ϕ, r ja r^2 sijoitetaan sopiviin ruutuihin esimerkiksi taulukon ylälaitaan. Maalien koordinaatit asetetaan omalle rivilleen vierekkäin niin, että x_1 tulee G-sarakkeeseen, y_1 H-sarakkeeseen jne. Laukausriveille kirjoitetaan A- ja B-sarakkeisiin @RAND-funktio, C- ja D-sarakkeisiin kaavojen (1) ja (2) mukaiset normaalijakautuneet satunnaisluvut (joihin siis U ja V haetaan A- ja B-sarakkeista), E- ja F-sarakkeisiin kaavojen (5) ja (6) mukaiset ampumasuunnan, iskemäkeskipisteen ja hajonnan huomioon ottavat satunnaisluvut eli iskemien x- ja y-koordinaatit X ja Y. G-sarakkeeseen sijoitetaan @IF-lause, joka tuottaa arvon 1, jos $(X - x_1)^2 + (Y - y_1)^2 < r^2$, ja muulloin arvon 0. Vastaavanlainen kaava kirjoitetaan jokaisen maalin x-koordinaatin alle. Ainakin yksi maali on saanut osuman, jos maalisarakkeiden alla on ainakin yksi ykkönen. Tätä voi testata sijoittamalla maalisarakkeiden oikealle puolelle, "osumasarakkeeseen", ehtolauseen, joka tuottaa ykkösen, jos maalisarakkeiden alla olevien ruutujen summa on positiivinen mutta nollan muulloin. Jos soluviittaukset soluihin, joissa on laukauksesta riippumatonta informaatiota (ammuntadata ja maalien koordinaatit) on muistettu tehdä kiinteisiin osoitteisiin ($-merkit), niin laukausrivin voi kopioida vaikka koko taulukon syvyyteen. Osumasarakkeessa olevien ykkösten summa jaettuna laukausrivien määrällä kertoo osumien osuuden ja on siis osumistodennäköisyyden estimaattori.

Harjoituksia

5. Rakenna osumasimulaattoriin luvun 2.4.2 mukaiset keskiarvo- ja keskihajontalaskurit ja määritä osumistodennäköisyydelle luotettavuusväli.

6. Rakenna kolmiulotteiseen maaliin, ilmamaaliin osumisen todennäköisyyden simulaatio. Neuvo: ilmamaalilla ja iskemällä on kolme koordinaattia.

3.1.2 Usean samanaikaisen laukauksen osumat

Useamman samanaikaisen laukauksen, esimerkiksi patteriston tai usean patteriston yhden tai usean yhteislaukauksen vaikutusta maaleihin (x_1, y_1), ..., (x_n, y_n) voi tarkastaa kirjoittamalla laukausrivejä sopiva määrä ja laskemalla sarakkeittain, onko jokin maaleista saanut osuman. Tämä tehdään laskemalla sarakesummia kunkin maalin kohdalta ja muodostamalla jokaisen maalin sarakkeeseen iskemärivien alapuolelle ehtolause, joka tuottaa ykkösen, jos sarakesumma on positiivinen ja muuten nollan. Kun nämä ykköset ja nollat lasketaan yhteen, saadaan simuloidussa tulituksessa tulleiden osumien määrä. Yksinkertaisin tilannehan on se, että esim. taistelijaan tullut yksi osuma on riittävän lamauttava, jolloin mahdollisesti tulevat lisäosumat eivät enää kasvata tulen tehoa.

Edellä esitetyssä esimerkissä oletettiin iskemäkeskipiste kiinteäksi. Tällöin kaikki vaihteluinformaatio sisältyi pituus- ja leveyshajontoihin σ ja τ. Jos ammutaan tuliyksiköllä, on kaikissa iskemissä mukana topografisen ja ballistisen valmistelun epätarkkuutta. Tämä voidaan ottaa huomioon satunnaistamalla iskemäkeskipiste (x_0, y_0) eli sijoittamalla ruudukon asianomaiseen ruutuun kaava, joka tuottaa oikealla tavalla hajontaa. Tuliyksikön kerrallisessa ammunnassa iskemäkeskipiste on kuitenkin aina sama, sen sijaan ammuntaa toistettaessa uudelleenlaskenta tuottaa uuden iskemäkeskipisteen. Näin päästään myös selvittämään kyseisten hajontatekijöiden merkitystä tulen osuvuuteen.

Edellä selostettua manuaalista uudelleenlaskentaa ja laskuritekniikkaa käyttäen ammunta voidaan suorittaa riittävän monta kertaa tilastollisen luotettavuuden saavuttamiseksi. Taulukkoon voi rakentaa laskurit osumia saaneiden maalien määrälle ja keskiarvolle sekä erikseen kullekin tuhoutuneiden maalien määrälle. – Laskenta, jossa toistettiin 1000 kertaa patteriston iskun (108 laukausta) tuottamat osumat neljälle maalille, kesti noin 35 sekuntia 2,6 GHz:n prosessorilla varustetulla pöytätietokoneella.

Harjoituksia

7. Rakenna osumasimulaattoriin tuliyksikön topografisen ja ballistisen valmistelun epävarmuuden simulointi.

3.2 Aluemaali

Aluemaaliin ammuttaessa iskemien generointi ei poikkea edellä esitellyistä pistemaalitapauksista, ei myöskään osumistodennäköisyyden laskeminen, kun osumien määrä on selvillä. Sen sijaan osumisen kriteeri on asetettava uudelleen. Osumisen kriteerinä ei nyt voi pitää satunnaisiskemän (X, Y) etäisyyttä kiinteästä pisteestä, vaan kuulumista maalin ja ampumatilanteen geometrian määrittämään pistejoukkoon.

Osumiskriteeri on suhteellisen helppo rakentaa silloin, kun maalialue on *kupera monikulmio* eli monikulmio, jonka kaikki kärjet ”osoittavat ulospäin”. ”Tavallisin” maalialuetyyppi, suorakaide, kuuluu tähän kategoriaan.

Olkoot monikulmion kärjet **vastapäivään** lueteltuina $V_1 = (x_1, y_1)$, $V_2 = (x_2, y_2)$, $\ldots$, $V_n = (x_n, y_n)$ ja olkoon vielä $(x_{n+1}, y_{n+1}) = V_{n+1} = V_1$. Pisteiden $V_k = (x_k, y_k)$ ja $V_{k+1} = (x_{k+1}, y_{k+1})$ kautta kulkevan suoran yhtälö xy-tasossa on

$$(x_{k+1} - x_k)(y - y_k) - (y_{k+1} - y_k)(x - x_k) = 0. \tag{7}$$

(Tämä suoran yhtälön muoto on monista erinäköisistä suoran yhtälön versioista käyttökelpoisin, koska se kattaa myös jommankumman koordinaattiakselin suuntaiset suorat.)

Suora jakaa tason kahdeksi puolitasoksi. Se, kumpaan näistä jokin suoraan kuulumaton tason piste (x, y) kuuluu, näkyy yhtälön vasemman puolen lausekkeen etumerkistä. Jos pisteet (x_k, y_k), (x_{k+1}, y_{k+1}) ja (x, y) ovat muodostamansa kolmion piirillä *positiivisessa järjestyksessä* eli niin, että kolmio pisteiden järjestyksessä kierrettäessä jää koko ajan vasemmalle puolelle, niin yhtälön (7) vasen puoli on positiivinen. Kuperan monikulmion sisäpiste on jokaisen piiriinsä kuuluvan janan määräämän suoran vasemmalla puolella, kun kierretään monikulmiota pisteiden numerojärjestyksessä. Niinpä monikulmion sisällä olevat pisteet (x, y) ovat sellaisia, joille yhtälön (7) vasen puoli on positiivinen kaikilla $k = 1, 2, \ldots, n$. Ulkopuolella olevat pisteet puolestaan ovat niitä, joille vasen puoli on negatiivinen ainakin yhdellä k:n arvolla.

Iskemän (X, Y) osumista maalialueen sisään voi siis testata laskemalla kutakin vierekkäistä maalialueen kärkipisteparia (x_k, y_k), (x_{k+1}, y_{k+1}) kohden suureen $(x_{k+1} - x_k)(Y - y_k) - (y_{k+1} - y_k)(X - x_k)$ ja muodostamalla ehtofunktion, jonka arvo on 1, kun suure on positiivinen, ja nolla muulloin. Jos näiden ehtofunktion arvojen tulo on 1 (tai vaihtoehtoisesti summa n), niin (X, Y) on maalialueella, muulloin ei. Käytännössä asia voidaan realisoida varaamalla jokaiselle maalialueen kärkipisteelle oma sarakkeensa, sijoittamalla pisteen x- ja y-koordinaatit tämän sarakkeen ylälaitaan ja laskemalla kutakin satunnaisiskemää kohden edellä mainitun ehtofunktion arvon kärkipisteen sarakkeeseen. Kullakin iskemärivillä lasketaan ehtofunktioiden nollien ja ykkösten tulo omaan sarakkeeseensa; jos tulo on yksi, laukaus on osuma. Sarakesumma antaa silloin suoraan osumien määrän, jota puolestaan voi käyttää osumistodennäköisyyden määrittämiseen.

Alueet, jotka eivät ole kuperia, mutta jotka voidaan kuitenkin rajata murtoviivalla (yleensä esimerkiksi panssarivaunun tai aluksen profiili), voidaan koota useammasta kuperasta monikulmiosta, joihin sitten sovelletaan edellä linjattua menettelyä. Osuma on sellainen laukaus, joka osuu yhteen osa-alueeseen.

Kaarevareunaisia alueita voidaan yleensä riittävällä tarkkuudella approksimoida monikulmioilla. Jos kaarevan reunan muoto tunnetaan jonkin yhtälön $f(x, y) = 0$ muodossa, lausekkeen $f(x, y)$ positiivisuutta tai negatiivisuutta on mahdollista käyttää hyväksi selvitettäessä jonkin pisteen kuulumista kuvioon eli sitä, onko satunnaisiskemä osuma vain ei.

Jos esimerkiksi bunkkerin profiili on ellipsin puolikas, bunkkerin leveys on 5 m ja korkeus 1,5 m, niin poikkileikkaustason piste (X, Y) on osuma, jos $Y > 0$ ja

$$\frac{X^2}{2{,}5^2} + \frac{Y^2}{1{,}5^2} < 1.$$

Harjoituksia

8. Simuloi osumista suorakaiteen muotoiseen maaliin, kun se on poikittain, pitkittäin tai vinossa asennossa ampumasuuntaan nähden.

9. Simuloi osumista erikokoisiin hajontaellipseihin (iskemäkeskipiste ellipsin keskipiste, puoliakselit esimerkiksi σ ja τ, 2σ ja 2τ, 3σ ja 3τ).

10. Rakenna monikulmio, joka kuvaa ajoneuvon profiilia tietystä suunnasta ja tietyltä etäisyydeltä tarkasteltuna. Simuloi osumista ajoneuvoa kohti ammuttaessa.

4 KAKSINTAISTELUN SIMULOINTI

Monet maailman ilmiöt voidaan mallintaa toisiaan seuraavina tiloina. Siirtyminen tilasta toiseen saattaa olla satunnaista, mutta kuitenkin jonkin todennäköisyyslain sääntelemää. Mahdollisilla tiloilla saattaa olla todennäköisyysjakauma. Sen tunteminen helpottaa tulevaisuuden ennustamista, vaikkei annakaan varmaa informaatiota asioiden kehittymisestä kussakin yksittäistapauksessa.

4.1 Puhdas tilatodennäköisyysmalli

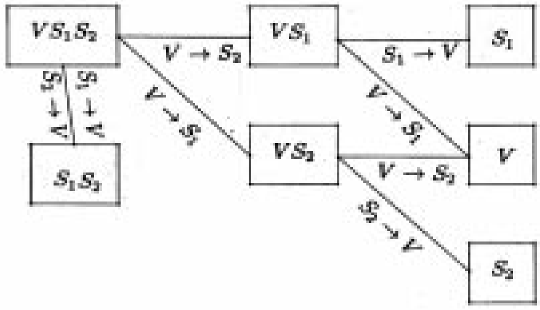
Jos systeemin eri tilojen välisten siirtymien todennäköisyydet ovat tiedossa, systeemin eri tiloihin päätymisen todennäköisyydet ovat laskettavissa. Jos siirtymistodennäköisyydet riippuvat vain tiloista, ei aikaisemmasta historiasta, puhutaan *Markovin ketjuista.*

Tarkastellaan esimerkkinä pelkistettyä kaksintaistelua, jossa toisena osapuolena on asejärjestelmä "V" (jonka voi hahmottaa panssarivaunuksi) ja toisella puolella kahdesta eri aseesta "S_1" ja "S_2" (jotka voi hahmottaa singoiksi) muodostuva asejärjestelmä. Ajatellaan vielä, että taistelu käydään erittäin kaavamaisesti: vuoronperään esiintyvät asevaikutukset, "laukaukset" $S_1 \to V$, $V \to S_1$, $S_2 \to V$ ja $V \to S_2$. Ammunnan seurauksena maali tuhoutuu todennäköisyyksin $P_{S_1 \to V}$, $P_{V \to S_1}$, $P_{S_2 \to V}$ ja $P_{V \to S_2}$. Taistelua jatketaan tietty kierrosmäärä tai siihen asti, kun toinen tai toinen osapuoli tuhoutuu.

V:n, S_1:n ja S_2:n muodostamalla järjestelmällä on kaikkiaan seitsemän tilaa, kun tilalla ymmärretään vielä tuhoutumatta olevien elementtien joukkoa. Tiloja voidaan merkitä itsensä selittävillä symboleilla VS_1S_2, VS_1, VS_2, S_1S_2, S_1, S_2 ja V. Tiloista kolme ensimmäistä on sellaisia, joissa taistelu jatkuu, kun taas neljä viimeistä tilaa ovat lopputiloja, joissa on enää jäljellä vain jommankumman osapuolen voimaa. Todennäköisyys, että järjestelmä on tilassa VS_1S_2, on $P_{VS_1S_2}$ jne.

Tarkasteltava kolmen aseen järjestelmä siirtyy tilasta VS_1S_2 tilaan S_1S_2, jos laukaus $S_1 \to V$ on tuhoava. Muussa tapauksessa systeemin tila on edelleen VS_1S_2. Samoin, jos systeemi on tilassa VS_1, niin tuhoava laukaus $S_1 \to V$ siirtää systeemin tilaan S_1. Vastaavalla tavalla voidaan analysoida muiden mahdollisten laukausten vaikutukset eri tilojen todennäköisyyksiin.

Taulukkolaskennassa tilojen todennäköisyyksien kehittyminen on verrattain yksinkertaisesti realisoitavissa. Laukausten tuhoamistodennäköisyydet sijoitetaan omiin ruutuihinsa. Kunkin tilan todennäköisyyksille varataan oma sarakkeensa. Kutakin laukausta varten kirjoitetaan rivi, jolle eri tilojen todennäköisyydet lasketaan suhteessa tilanteeseen ennen kyseistä laukausta. Tämä



Kaksintaistelun tilat

tarkoittaa, että kunkin rivin todennäköisyydet lasketaan kaavoilla, joissa syötteinä ovat edellisen rivin todennäköisyydet ja eri laukausten tuhoamistodennäköisyyksistä johtuvat siirtymistodennäköisyydet. Ylimmällä rivillä on tilan VS_1S_2 todennäköisyytenä 1 ja muiden tilojen todennäköisyytenä 0. Sellaisten tilojen, jotka eivät voi muuttua tietyssä laukauksessa, todennäköisyys kopioidaan suoraan edelliseltä riviltä. Esimerkiksi laukaus $S_1 \to V$ ei muuta tilojen VS_2, S_2 ja V todennäköisyyksiä. Sen sijaan $P_{VS_1S_2}$ muuttuu todennäköisyydeksi $P_{VS_1S_2} \cdot (1 - P_{S_1\to V})$ ja $P_{S_1S_2}$ lisääntyy määrällä $P_{VS_1S_2} \cdot P_{S_1\to V}$ Vastaavasti laukaus $V \to S_1$ lisää tilan VS_2 todennäköisyyttä määrällä $P_{VS_1S_2} \cdot P_{V\to S_1}$ ja tilan V todennäköisyyttä määrällä $P_{VS_1} \cdot P_{V\to S_1}$ ja pienentää tilan VS_1S_2 ja VS_1 todennäköisyyksiä kertoimella $1 - P_{V\to S_1}$. Sen sijaan tilojen S_1S_2, S_1 ja S_2 todennäköisyyksiin laukaus $V \to S_1$ ei vaikuta.

Kutakin laukaustyyppiä vastaavat todennäköisyyksien muutosrivit on helppo kirjoittaa. Erityyppisten laukausten sijoittaminen jonoon on nyt hoidettavissa kopioi–liitä toiminnalla. Muutamien rivien kopioimisen jälkeen alkaa näkyä, että tilojen, joissa taistelua vielä voisi jatkaa, todennäköisyydet käyvät erittäin pieniksi ja lähes koko todennäköisyysmassa keskittyy lopputiloihin. Lopputilojen todennäköisyysjakauma heijastelee kaksintaistelun todennäköistä lopputulosta. Esitetyssä kehyksessä se on melko herkkä paitsi tuhoamistodennäköisyyksille myös tulenavausjärjestykselle.

4.2 Kaksintaistelu, jossa ammuntajärjestys on satunnainen

Edellinen tilatodennäköisyyksien lasku ei ollut varsinaisesti simulointia, vaan pikemminkin eräs todennäköisyyslaskennan menetelmä. Asiaa voi lähteä tarkastelemaan myös niin, että seurataan tilojen kehittymistä satunnaisesti.

Voidaan esimerkiksi valita kulloinkin tulittava ase sattumanvaraisesti (vielä mukana olevista). Nimetään järjestelmän mahdolliset tilat numeroin, esimerkiksi $VS_1S_2 = 1$, $VS_1 = 2$, $VS_2 = 3$, $V = 4$, $S_1S_2 = 5$, $S_1 = 6$ ja $S_2 = 7$. Erityyppiset laukaukset numeroidaan myös, esimerkiksi $V \to S_1 = 1$, $V \to S_2 = 2$, $S_1 \to V = 3$ ja $S_2 \to V = 4$. Varataan taulukkoon kaksi saraketta, toinen järjestelmän tilalle ja toinen laukaukselle, joka kyseisen tilan vallitessa ammutaan ja joka siis voi muuttaa tilaa. Jos tilamuuttujan arvo on 4 tai enemmän, taistelu on päättynyt. Tällöin laukaussarakkeeseen tuotetaan 0. Generoidaan joka riville kaksi satunnaislukua. Ensimmäisen perusteella valitaan, minkä typpinen laukaus ammutaan. Jos tila on 1, mahdollisia laukauksia on 4. tällöin laukauksen numero voi olla suoraan 1+@INT(4∗@RAND), jos kaikki mahdollisuuden valitaan yhtä todennäköisiksi. Jos tila on 2 tai 3, mahdollisia laukauksia on vain kaksi: tilassa 2 $V \to S_1$ eli 1 ja $S_1 \to V$ eli 3 ja tilassa 3 $V \to S_2$ eli 2 ja $S_2 \to V$.

Sijoitetaan kaksi satunnaislukua sarakkeisiin A ja B. Olkoon tilamuuttuja sarakkeessa C ja laukausmuuttuja sarakkeessa D. Kun käytetään luvussa 2.3.2 esiteltyä diskreetin tasajakauman menetelmää, voidaan esimerkiksi rivin 8 laukaussarakkeeseen, siis ruutuun D8, kirjoittaa kaava @IF(C8 = 1; 1+@INT(4∗A8); @IF(C8 = 2; 1 + 2∗@INT(2∗A8); @IF(C8 = 3; 2 + 2∗@INT(2∗@RAND);0))). Tilanmuutos riippuu edellisestä tilasta, ammuttavasta laukauksesta ja siitä tuhoaako laukaus. Jos laukauksien tuhoamistodennäköisyydet ovat ruuduissa seuraavasti: B1: $P_{V\to S_1}$, B2: $P_{V\to S_2}$, B3: $P_{S_1\to V}$ ja B4: $P_{S_2\to V}$, niin tilasarakkeeseen riville 9, siis ruutuun C9, voidaan sijoittaa ensi silmäykseltä mutkikkaan näköinen useampitasoinen ehtolause

@IF(C8 = 1; @IF(D8 = 1; @IF(B8 < $B$1; 3; 1); @IF(D8 = 2; @IF(B8 < $B$2; 2; 1); @IF(D8 = 3; @IF(B8 < $B$3; 5; 1); @IF(B8 < $B$4; 5; 1)))); @IF(C8 = 2; @IF(D8 = 1; @IF(B8 < $B$1; 4; 2); @IF(B8 < $B$3; 6; 2)); @IF(C8 = 3; @IF(D8 = 2; @IF(B8 < $B$2; 4; 3); @IF(B8 < $B$4; 7; 3)); C8)))

Siinä käydään läpi se, onko edellinen tila, siis ruudun C8 arvo, 1, 2, 3 vai suurempi kuin 3, ja vaihtoehtojen 1, 2 ja 3 tapauksissa se, mikä laukaus ammuttiin (ruudun D8 arvo) ja se, oliko laukaus tuhoava (ruudussa B8 olevan satunnaisluvun arvon vertailu ruuduissa B1 – B4 oleviin kriteeriarvoihin).

Tila- ja laukaussarakkeisiin voi kopiointitoiminnalla synnyttää pari- kolmekymmentä riviä. Mitä suuremmat tuhoamistodennäköisyydet, sitä nopeammin tila stabiloituu yhdeksi lopputiloista $4-7$ ja laukausmuuttujan arvoksi tulee 0. Alimman tilarivin tuloksista voi rakentaa laskurin, joka laskee erilaisten lopputulosten jakaumaa. Taistelun kestoa voi tutkia vaikkapa laukaussarakkeen viereen synnytetyillä laskureilla, jotka reagoivat tilanteeseen, jossa viimeisen kerran esiintyy nollaa suurempi laukausmuuttujan arvo.

Harjoituksia

11. Edellisessä mallissa laukausten määrää ei ole rajoitettu. Rakenna malliin lisäehdot, jotka ottavat huomioon osapuolten laukausten maksimimäärän.

5 TAPAHTUMAOHJATTUA SIMULOINTIA

Tapahtumaohjattu simulointi sisältää aikatekijän. Tarkasteltavaan ilmiöön liittyvät parametrit muuttuvat *tapahtumahetkinä*, joiden väliajat ovat satunnaistettuja. Parametrien muutokset ovat nekin satunnaistettuja. Simulaatio reaalistuu tapahtumaluettelona. Tyypillinen tapahtumaohjatun simuloinnin esimerkki on *jono*.

5.1 Jonon simulointi

Monet toiminnat voidaan kuvata palvelija–asiakas-mallilla, johon liittyy jono. Toimintaa suorittava palvelija voi käsitellä vain tiettyä määrää asiakkaita kerrallaan. Jos palvelija on varattu, palvelua tarvitseva asiakas asettuu jonoon. Kun palvelija saa asiakkaan tehtävänsä edellisen asiakkaan kanssa suoritetuksi, se ottaa jonosta uuden asiakkaan. Jonon pituus voi olla rajoitettu, palvelijoita voi olla useita tai sama asiakas vaatii peräkkäin usean eri palvelijan palveluja. Jos palvelu–jonotus tilanne jatkuu pitkään, voidaan usein puhtaasti matemaattisen analyysin keinoin määrittää jonon keskimääräinen pituus, keskimääräinen jonotusaika ja palvelijan teho [1, s. 59 – 69].

Palvelija–asiakas-malli on tietysti sovellettavissa varsinaiseen asiakaspalveluun kuten marketin lihatiskille, mutta myös esim. korjaamon toimintaan, atk-palvelimeen, viestikeskukseen tai tulitukiyksikköön.

Simuloidaan yksinkertaista tilannetta, jossa palvelija palvelee yhtä asiakasta kerrallaan. Asiakkaiden saapuminen palveluun on sattumanvaraista. Saapumisaikojen väliaika noudattaa tällöin eksponettijakaumaa, jonka parametri a osoittaa asiakkaiden keskimääräisen luvun aikayksikköä kohden. Palveluaika voi olla vakioitu tai se voi olla satunnainen. Palvelun laadusta riippuen palveluajan jakauma saattaa olla hyvin erilainen: esimerkkeinä voisi ajatella tasajakaumaa jollekin aikavälille, normaalijakaumaa jonkin keskimääräisen ajan ympärille tai eksponenttijakaumaa noudattava.

Mallissamme on palvelija ja rajoittamaton jono. Asiakkaat saapuvat yksitellen. Tapahtumia voi olla neljä erilaista. Numeroidaan ne seuraavasti

1 Asiakas saapuu ja pääsee heti palveluun

2 Asiakas saapuu, mutta palvelija on varattu. Asiakas jää jonoon.

3 Palvelija saa työnsä valmiiksi, ja ottaa seuraavan asiakkaan jonosta.

4 Palvelija saa työnsä valmiiksi. Jonossa ei ole ketään, ja palvelija odottaa seuraavaa asiakasta.

Kutakin tapahtumaa vastaa taulukkolaskentamallissamme rivi. Tapahtumalajille varataan taulukossa oma sarake. Jonossa olevien asiakkaiden määrää, siis jonon tilaa, ylläpidetään omassa sarakkeessaan. Ajan kulkua on mukava seurata kolmessa sarakkeessa. Näistä yhdessä ovat tapahtumien ajat, yhdessä pidetään kirjaa asiakkaiden saapumisajoista ja yhdessä palvelujen päättymishetkistä.

Kun tapahtuma on 1 tai 2, tuloaikasarakkeen lukua lisätään eksponenttijakautuneen satunnaisluvun verran, siis määrällä $-a*$@LN(@RAND). Jos tapahtuma on 3 tai 4, sarakkeen luku luetaan sellaisenaan edelliseltä riviltä. Tuloaikasarakkeessa on siten aina seuraavan asiakkaan saapumisaika. Palvelun päättymisaikasarakkeen luvun arvo saadaan tiloissa 1 ja 3 lisäämällä tapahtuma-aikaan palveluaika. Tilassa 2 palvelun valmistumisaika luetaan suoraan edelliseltä riviltä. Tilassa 4 valmistumisajaksi on syytä asettaa seuraavan asiakkaan tuloaika. Palveluaika voidaan asettaa vakioksi tai sitten palvelulle generoidaan satunnainen pituus. Pituuden ilmoittava satunnaisluku voidaan tietysti rakentaa omassa sarakkeessaankin.

Kolmas aikasarake sisältää tapahtumahetkien luettelon. Siihen ohjelmoidaan edellisen rivin tuloaika- ja palvelunpäättymisaikasarakkeista pienempi luku.

Itse tapahtumasarakkeessa otetaan huomioon aikasarakkeiden sisältämä tieto. Jos systeemin tila on 4, seuraava tapahtuma voi olla vain 1. Jos systeemin tila on muu kuin 4, niin seuraava tapahtuma on 2, jos seuraava tuloaika on seuraavaa valmistumisaikaa pienempi. Jos taas seuraava valmistumisaika on pienempi, seuraava tapahtuma on 4, jos jonossa ei ole ketään, mutta 3, jos jonoa on. Oletetaan, että tapahtumasarake on A, tuloaikasarake C, valmistumisaikasarake D ja sarake, jossa kerrotaan jonon pituus, G. Silloin esimerkiksi ruudussa A6 voisi olla ehtolause

$$\text{@IF}(\text{A5} = 4;\ 1;\ \text{@IF}(\text{C5} < \text{D5};\ 2;\ \text{@IF}(\text{G5} > 0;\ 3;\ 4))).$$

Jonon pituuden ilmaisevassa sarakkeessa on luku, jota lisätään aina yhdellä, kun tapahtuma on 2, vähennetään yhdellä, kun tapahtuma on 3. Tapahtumissa 1 ja 4 jonosarakkeen luku on suoraan edellisen rivin luku.

Taulukkoon voidaan helposti liittää laskureita, jotka kertovat jo palveltujen asiakkaiden lukumäärän, jonon maksimipituuden, jonotukseen kuluvan ajan jne. Esimerkiksi jonotukseen kaikkiaan kuluva aika saadaan laskemalla joka rivillä jonon pituuden ja seuraavan ja kyseisen rivin tapahtuma-aikasarakkeen arvo. Näiden summa kertoo jonottajien kokonaisjonotusajan.

Harjoituksia

12. Rakenna jonopalvelumalli, jossa palvelijan käyttämä työaika on eksponenttijakautunut ja toinen, jossa se on normaalijakautunut (ota huomioon, että normaalijakautunut satunnaisluku voi olla negatiivinen, ja estä tällaiset palveluajat sopivalla ehtolauseella).

5.1.1 Muunnelmia

Jonon perusmallia voi helposti varioida. Yhden palvelijan sijasta voi käyttää useita eri palvelijoita ja näiden työhön käyttämän ajan jakauma voi vaihdella. Asiakas voi valita eri palvelijoista tietyn sattumanvaraisesti tai valintaa voi säädellä ennalta määrätty järjestys. Kunkin palvelijan työn valmistumisaika voidaan sijoittaa omaan sarakkeeseensa. Simulointiin voi liittää laskurin, joka pitää kirjaa siitä, kuinka suuri osa palvelutehosta on käytössä.

Jonolla voi olla maksimipituus. Teknisistä syistä viestinvälitin ei voi pitää kuin tietynmittaista jonoa ja parturin odotushuoneen tuolien lukumäärä on rajallinen. Simuloinnin tapahtumavalikoimaan liitetään tapahtuma "asiakas saapuu, mutta ei pääse jonoon". Tällainen tapahtuma ei muuta rekistereitä, mutta on tietysti palveluntarjoajan ja asiakkaan kannalta ei-toivottava, joten sen yleisyyden arvioiminen simuloinnilla voi olla mielekästä.

Harjoituksia

13. Rakenna malleja, joissa jonon pituus on rajattu. Tutki, kuinka paljon palvelun tehokkuuteen vaikuttaa tämä rajaus ja se, kuinka ankara se on.

5.2 Vikaantumisen simulointia

Luotettavuusteorian oppien mukaan järjestelmä vikaantuu normaalina käyttöaikanaan satunnaisesti niin, että aika ennen seuraavaa vikaantumista on eksponenttijakautumaa noudattava satunnaisluku. Vikaantunut järjestelmä on poissa käytöstä sen korjaamisen vaatiman ajan tai ajan, jonka vaatii uuden järjestelmän hankkiminen ja käyttöön saaminen.

Rakennetaan yksinkertainen malli kolmesta yhtä aikaa käytettävästä järjestelmästä. Kullakin on oma vikataajuutensa tai keskimääräinen vikaantumisien välinen aika. Kunkin järjestelmän korjaaminen käyttökuntoon kestää keskimäärin tietyn ajan. Tämä aika on normaalijakautunut tietyllä odotusarvolla ja keskihajonnalla.

Sijoitetaan nämä aikaparametrit kolmen sarakkeen ylälaitaan, esimerkiksi ruutuihin B1, D1 ja F1 keskimääräiset vikaantumisien väliset ajat, ruutuihin B2, D2 ja F2 keskimääräiset korjausajat ja ruutuihin B3, D3 ja F3 korjausaikojen keskihajonnat. Varataan sarakkeet A, C ja D sen osoittamiseen, onko kukin järjestelmistä käytössä vai ei. Käytössä olo osoitetaan 1:llä, vikaantuneena oleminen 0:lla. Toiminnan alkaessa kaikki kolme järjestelmää toimivat. Osoitetaan tämä sijoittamalla esimerkiksi ruutuihin A4, C4 ja E4 ykkönen. Sarakkeisiin B, D ja F sijoitetaan kunkin järjestelmän kannalta seuraava tapahtuma-aika. Tapahtuma voi olla vikaantuminen tai korjauksen jälkeen käyttöön palautuminen. Ensimmäinen tapahtuma jokaiselle järjestelmälle on vikaantuminen. Jos vikaantumishetki on eksponenttijakautunut, niin ensimmäisen järjestelmän vikaantumishetki saadaan ruutuun B4 kaavalla −B$1∗@LN(@RAND) ja ruutuihin D4 ja F4 kopioimalla niihin ruudun B4 kaava. Ensimmäinen tapahtumahetki on pienin ruutujen B4, D4 ja F4 luvuista. Sijoitetaan tämä ruutuun G4 kaavalla @MIN(B4;D4;F4). Seuraavalle riville ruutuun A5 kirjoitetaan kaava @IF($G4 =B4;1−A4;A4) ja kopioidaan se ruutuihin C4 ja E4. Kaava muuttaa sen järjestelmän, jota tapahtuma-aika koskee, tilan päinvastaiseksi, ensi vaiheessa siis 1:stä 0:ksi. Muiden järjestelmien tilamuuttujat pysyvät samoina.

Ruutuun B5 kirjoitetaan ajan kasvamista osoittava kaava. Kaava on @IF(A4−A5 = 0;B4; @IF(A4−A5 = 1;B4+@NORMAL(@RAND;B$2;B$3;1); B4−B$1*@LN(@RAND))). Jos systeemin tila ei ole muuttunut, seuraava relevantti aika on yläpuolisen rivin aika. Jos systeemin tilanmuuton on ollut vikaantuminen, seuraava relevantti aika on korjauksen valmistuminen. Jos tilan muutos on ollut korjauksen valmistuminen, seuraava relevantti aika on uusi vikaantumisaika. Kopioidaan ruudun B5 kaava ruutuihin D5 ja F5. Kun vielä ruudun G4 minimikaava kopioidaan ruutuun G5, ollaan valmiita koko rivin A5..G5 kopioimiseen alla oleville riveille. Simuloitua tietoa voidaan kerätä esimerkiksi H4:ään kirjoitettavalla kaavalla A4+C4+F4, joka kertoo, kuinka moni järjestelmistä on kulloinkin käytössä, ruutuun I4 kirjoitettavalla kaavalla H4∗(G4-G3), ja J4:n kaavalla @SUM($I$4..I4)/(3∗G4). Kun riviä H4..J4 kopioidaan eri riveille alapuolelle, saadaan osoittimet käytössä olevien järjestelmien määrälle, tehokkaalle työajalle ja tämän osuudelle maksimityöajasta.

Harjoituksia

14. Muuta edellistä simulointia niin, että järjestelmän korjausaika on a) vakio, b) eksponenttijakautunut.

15. Lisää edelliseen simulointiin laskurit, jotka pitävät kirjaa siitä, kuinka pitkän ajan kunnossa on k järjestelmää, k:n arvoilla 0, 1, 2 ja 3.

6 VERKON YHTENÄISYYDEN SIMULOINTI

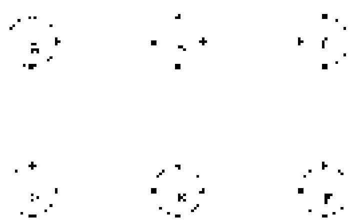
Monia rakenteita, kuten tiestöä tai viestiverkkoa voidaan kuvata *verkoksi* nimitetyllä matemaattisella rakenteella. Se koostuu *solmuista* ja joitakin näistä yhdistävistä *sivuista*. Joissain tilanteissa on hyödyllistä liittää sivuun suunta (kuten silloin, kun tieosuudet on säädetty yksisuuntaisiksi). Käsitellään tässä vain yksinkertaista tapausta, jossa kunkin kahden solmun A ja B välissä joko on tai ei ole yhteys, ja suuntaa ei oteta huomioon.

Jos verkon jokaisesta solmusta on mahdollisuus kulkea yhteysvälejä pitkin mahdollisesti muiden solmujen kautta jokaiseen verkon solmuun, verkko on *yhtenäinen*. Kysymys verkon yhtenäisyydestä tulee vastaan esimerkiksi kun viesti-, tie- tai voimansiirtoverkko tulee joltain osiltaan käyttökelvottomaksi. Jos yhteysvälejä on paljon, niistä muutamien poistaminen ei ehkä vielä tuhoa yhtenäisyyttä.

6.1 Verkon matriisiesitys

Verkko on sinänsä geometris-visuaalinen käsite. Tietokoneelle se on voitava esittää numeerisessa muodossa.

Verkkoa on taulukkolaskennan kannalta mukava esittää *yhteysmatriisina* M, jossa kutakin solmua vastaa sekä vaakarivi että pystysarake ja solmujen A ja B välisen yhteyden eli sivun AB olemassaolo merkitään asettamalla 1 sekä solmua A vastaavan rivin ja solmua B vastaavan sarakkeen että solmua B vastaavan rivin ja solmua A vastaavan sarakkeen risteyskohtaan. Sellaisia solmupareja, joiden välillä ei ole sivua, vastaavat risteyskohdat merkitään 0:lla. Esimerkiksi oheisen kuvan mukaisen verkon yhteysmatriisi olisi



$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}. \tag{8}$$

Matriisien $M = (a_{ij})$, i, $j = 1, 2, \ldots, n$, ja $N = (b_{jk})$, j, $k = 1, 2, \ldots, n$ tulo

MN on määritelmän mukaan se matriisi, jonka i:nnen rivin k:s alkio on

$$c_{ik} = a_{i1}b_{1k} + a_{i2}b_{2k} + \cdots + a_{in}b_{nk}. \tag{9}$$

Jos verkon i:nnestä solmusta on yhteys j:nteen ja j:nnestä k:nteen, niin verkossa on yhteys i:nnestä solmusta j:nnen kautta k:nteen. Yhteysmatriisin i:nnen rivin j:nnessä sarakkeessa ja j:nnen rivin k:nnessa sarakkeessa on tällöin 1. Silloin tulomatriisin $M \cdot M = M^2$ i:nnen rivin ja k:nnen sarakkeen risteyksessä oleva luku on positiivinen. Matriisin M^2 nollasta eroavat alkiot osoittavat kaikki yhteydet, joiden pituus on kaksi solmunväliä. Vastaavasti $M^3 = M \cdot M^2$ sisältää ne yhteydet, joiden pituus on 3 solmunväliä. Jos solmuja on n, pisin yhteys on $n-1$:n solmuvälin mittainen. Koska matriisit lasketaan yhteen alkioittain, matriisin

$$N = M + M^2 + \cdots + M^{n-1}$$

positiiviset alkiot osoittavat ne solmuparit, joiden välillä on yhteys.

6.1.1 Matriisitulo ja matriisin potenssit taulukkolaskennassa

Matriisien kertominen on taulukkolaskentaohjelmissa valikoista löytyvänä valmistoimintana. Monte Carlo -simulointia varten on kuitenkin käytännöllisempää toteuttaa kertolasku laskukaavana. Kopiointitoiminnan ansiosta kaavaa (9) ei kuitenkaan tarvitse kirjoittaa kuin kerran. Oletetaan, että matriisi A on taulukon riveillä 1, 2, 3 ja 4, sarakkeissa A, B, C ja D eli alueessa A1..D4. Jos matriisi B on riveillä 5, 6, 7 ja 8, sarakkeissa A, B, C ja D, eli alueessa A5..D8, niin matriisitulo BA tuotetaan kirjoittamalla ruutuun A9 kaava $A5*A$1+$B5∗A$2+$C5∗A$3+$D5∗A$4 ja kopioimalla tämä alueeseen A9..D12.

Jos matriisi B on sama kuin matriisi A, niin alueeseen A9..D12 syntyvä matriisi on A^2. Jos edellä kopioitu kaava liitetään ruutuun A13, siihen ilmestyy $A9∗A$1+$B9∗A$2+$C9∗A$3+$D9∗A$4, eli tulomatriisin $(BA)A = BA^2$ vasemman yläkulman alkio. Kun kaava liitetään koko alueeseen A13..D16, siihen syntyy matriisi BA^2 kokonaisuudessaan. Jos $B = A$, alueessa on matriisi A^3. Näin havaitaan mielenkiintoinen seikka: kaikki matriisin A potenssit voidaan tuottaa pelkästään kopioimalla yhtä kaavaa.

6.2 Haavoittuva verkko

Taulukkolaskentasimuloinnilla voidaan luoda kuvaa yksinkertaisen verkon rakenteen vaikutuksesta sen haavoittuvuuteen. Käsitellään esimerkkinä edellisen kuvan verkkoa. Oletetaan, että jonain hetkenä kukin verkon sivu on ehjä todennäköisyydellä p, ja että eri välien eheys on toisista väleistä riippumaton. Tällaista verkkoa voidaan kuvata yhteysmatriisilla M, missä matriisin (8) ykköset on korvattu satunnaismuuttujilla, joiden arvo on 1 todennäköisyydellä p ja 0 todennäköisyydellä $1 - p$. On kuitenkin huomattava, että kukin matriisin päälävistäjän suhteen symmetrinen ykkönen edustaa samaa väliä, joten tällaisissa asemissa olevat satunnaismuuttujat ovat samat. Pyritään selvittämään, miten todennäköisesti tällaisessa verkossa on yhteys jonkin kahden solmun välillä ja miten todennäköisesti verkko on yhtenäinen, eli sellainen, jossa jokaisesta solmusta on yhteys jokaiseen muuhun solmuun.

Taulukkolaskentaan tällainen matriisi voidaan tuottaa sijoittamalla jonkin 6×6 neliön oikeaan yläkulmaan puuttuvat välit nollina ja epävarmat välit kaavoina @IF(@RAND< p; 1; 0). p on sijoitettu johonkin ruutuun, johon viitataan dollarimerkein kiinnitetyllä osoitteella. Muodostetaan tästä yläkulmasta symmetrinen yhteysmatriisi suorin soluviittauksin. Tämä tapahtuu muutamin kopioi–liitä-komennoin. Yhteysmatriisin potenssit toisesta viidenteen lasketaan edellä kuvatulla tavalla yhteysmatriisin alle. Potenssisummamatriisi $S = M + M^2 + \cdots + M^5$ muodostetaan kirjoittamalla sen vasemman yläkulman alkion summakaava ja kopioimalla tämä kaava edelleen 6×6 alueeseen. Kun olemme kiinnostuneita siitä, onko tiettyjen solmujen välillä yhteyttä, voimme sijoittaa S-matriisin alkioiksi @IF-kaavan avulla ykkösen, jos potenssien summassa alkio on positiivinen. Jos S:n kaikki alkiot ovat ykkösiä, jokaisesta solmusta on yhteys jokaiseen muuhun. Itseviittaustekniikalla on edelleen helppo rakentaa laskurit, joiden avulla voi laskea eri solmuvälien yhteyksien lukumäärän, kun arvonta toistetaan useamman kerran. Kun todennäköisyydeksi p asetettiin 0,9 ja yhteydet arvottiin 1000 kertaa, saatiin eri yhteyksien frekvenssimatriisiksi

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 952 & 918 & 856 & 922 & 905 & 853 \\ 918 & 993 & 924 & 922 & 963 & 914 \\ 856 & 924 & 967 & 864 & 909 & 931 \\ 922 & 922 & 864 & 967 & 932 & 872 \\ 905 & 963 & 909 & 932 & 997 & 932 \\ 853 & 914 & 931 & 872 & 932 & 974 \end{pmatrix} \end{array}$$

Aikaa kului noin 35 sekuntia. Päälävistäjän alkiot osoittavat niiden tapauksien lukumäärän, joissa solmusta ylimalkaan on yhteys johonkin toiseen solmuun. Esimerkkitapauksessa verkko oli yhtenäinen 802 kertaa. Tässä tapauksessa verkon yhtenäisyyden todennäköisyys asettui lähelle yksittäisen sivun eheyden todennäköisyyttä. Kun simulointi toistettiin arvolla $p = 0,9$, verkko oli yhtenäinen 942 kertaa ja yhteys A:sta F:ään syntyi 958 kertaa.

Harjoituksia

16. Selvitä, mitä yhteyksien lisääminen, esimerkiksi välille AE ja BF, vaikuttaa edellä käsitellyn verkon haavoittuvuuteen. Selvitä, mitä vaikutusta yhteyden BE poistamisella olisi.

17. Tutki, miten yhteysvälien kahdentaminen tai kolmintaminen (vikaantumisen mielessä toisistaan riippumattomin rinnakkaisvälein) vaikuttaa verkon yhtenäisyyteen. Vihje: Mieti, miten todennäköistä on, että kaikki yhteydet ovat yhtä aikaa poikki vertaa arvottua satunnaislukua tähän todennäköisyyteen.

7 JOUKKOJEN KULUMISEN SIMULOINTI

7.1 Lanchesterin deterministiset kulumismallit

Englantilaisen *F. W. Lanchesterin* ensimmäisen maailmansodan aikana esittämä kuuluisa joukkojen kulumisen malli on edelleen monien taistelun kvantitatiivisten analyysien pohjana. Lanchesterin [1, s. 10] perusmalleissa taistelevan joukon vahvuus pienenee nopeudella, joka joko on verrannollinen vastustajan vahvuuteen tai omaan ja vastustajan vahvuuteen. Täten molempien taistelun osapuolten vahvuuden muutokset kytkeytyvät toisiinsa, ja niitä on tarkasteltava yhdessä. Jos kulumistapa on molemmin puolin sama, puhutaan edellisessä tapauksessa tähdätyn tulen mallista ja Lanchesterin neliölaista, jälkimmäisessä aluevaikutuksen mallista ja Lanchesterin lineaarisesta laista. Toisiaan kuluttavat joukot voidaan jakaa osiin, joiden kulutus ja kuluminen suhteessa vastustajan osiin on erilaista.

Lanchesterin malli käsittelee joukon vahvuutta jatkuvana suureena. Malli voidaan muotoilla differentiaaliyhtälöryhminä, jotka yksinkertaisissa tapauksissa voidaan ratkaista suljetussa muodossa ja jotka aina voidaan likimäärin ratkaista numeerisin menetelmin. Perusidea, muutos, jonka suuruus kulloinkin määräytyy osapuolten vahvuuksista, on kuitenkin helposti simuloitavissa taulukossa diskretisoidussa muodossaan. Totuuden nimessä on muistettava, että nyt simuloidaan mallia, joka yrittää simuloida todellisuutta; taistelu on itsessään niin monielementtinen asia, että sen rehellinen simulointi on jokseenkin mahdotonta.

7.1.1 Homogeeniset joukot

Yksinkertaisimmillaan Lanchesterin malli on neliölaissa. Siinä osapuolten S ("Sininen") ja K ("Keltainen") vahvuudet N_S ja N_K muuttuvat aikana Δt määrillä

$$\Delta N_S = -a_K N_K \Delta t \quad \text{ja} \quad \Delta N_K = -a_S N_S \Delta t.$$

Taistelevien osapuolien alkuvahvuudet sijoitetaan kahteen sarakkeeseen, esimerkiksi ruutuihin A1 ja B1, joukon tulinopeuden ja tehon sekä vastustajan suojautumisasteen huomioon ottavat kertoimet a_K ja a_S kiinteisiin ruutuihin, esimerkiksi C1 ja D1. Mukavuuden vuoksi voi ajan yksikön valita niin, että $\Delta t = 1$. Kirjoitetaan ruutuihin A2 ja B2 kaavat +A1−\$C\$1∗B1 ja +B1−\$D\$1∗A1 ja kopioidaan näitä kaavoja sarakkeisiin A ja B.

Taistelua voidaan käydä tietty aika, mikä merkitsee kiinteätä määrää aika-askelia eli taulukkolaskennan rivejä. Voittanut osapuoli on se, jonka vahvuus tällä rivillä on suurempi. Lanchesterin malleissa taistelun katsotaan usein päättyvän silloin, kun jommankumman osapuolen vahvuus laskee alle jonkin kynnysarvon, esimerkiksi 70 % alkuvahvuudesta. Tämä osapuoli on samalla hävinnyt osapuoli. Yksinkertainen keino nähdä voittaja eri lähtöparametrien arvoilla on sijoittaa vahvuusarvojen viereen ehtolauseet, jotka esimerkiksi tuottavat kahteen sarakkeeseen joko tyhjän merkin tai ykkösen sen mukaan, onko vahvuus alittanut asetetun kynnyksen. Laskemalla sarakkeiden summa eli ykkösten määrä saadaan selville, kumpi osapuoli aikaisemmin alitti kynnysvahvuuden ja hävisi taistelun.

Aluevaikutteisen tulen käsittely on muuten sama, mutta A2- ja B2-ruutujen kaavoiksi on kirjoitettava

+A1−$C$1∗B1∗A1/$A$1 ja +B1−$D$1∗A1∗B1/$B$1.

Kaavoihin sisältyvän alkuvahvuuksilla jakamisen voisi kuitata myös pienentämällä (huomattavasti!) kulutusparametreja a_K ja a_S, mutta esitetty kaava noudattaa Lanchesterin teoriassa totuttua muotoilua.

Taulukkoon voidaan liittää ruutuja, jossa osapuolen voimaa kasvatetaan taisteluun mukaan tulevan reservin voimalla.

7.1.2 Epähomogeeniset joukot

Jos Lanchesterin mallia yritetään sovittaa joukkoihin, jotka ovat jakautuneet selvästi erilaisiin komponentteihin (esim. jalkaväki, panssarit, tykistö), on otettava huomioon toisaalta tulen erilainen vaikutus ja tulen jakautuminen eri kohteisiin. Lisäksi osa tulesta voi olla aluevaikutteista ja osa tähdättyä.

Jos Sininen jakautuu k:hon ja Keltainen n:ään osaan, tehoparametreja $a_{S_{ij}}$, $b_{K_{ji}}$ tulee $2kn$ kappaletta (jokaisen kummankin puolen osion vaikutus kuhunkin toisen puolen osioon), joista jotkut voivat olla nollia. Lisäksi kuvaan kuuluu $2kn$ osuusparametria, joilla osoitetaan kunkin se osa kunkin osapuolen voimasta, joka kohdistuu tiettyyn vastustajan osaan. Osuusparametri ja vastaava tehoparametri voidaan kertoa keskenään osuuden ja voiman osoittavaksi kertoimeksi (jota seuraavassa edelleen merkitään $a_{S_{ij}}$:llä ja $b_{K_{ji}}$:llä. Tietyn Sinisen osion S_i voiman muutos on aikajaksona Δt on tähdätyn tulen mallissa

$$\Delta N_{S_i} = -\left(b_{K_{1i}} N_{K_1} + \cdots + b_{K_{ni}} N_{K_n}\right) \Delta t.$$

Vastaavanlainen on tietyn Keltaisen osion K_j voiman muutos:

$$\Delta N_{K_j} = -\left(a_{S_{1j}} N_{S_1} + \cdots + a_{S_{kj}} N_{S_k}\right) \Delta t.$$

Taistelun kulun seuraamiseksi eri osioiden vahvuuksille varataan sarake kullekin, vaikutus- ja osuusparametreille omat alueensa. Näiden tulo voidaan muodostaa yhden kaavan kopioinnilla. Vahvuussarakkeiden rivit generoidaan ylemmistä riveistä kulumiskaavojen avulla (kiinnitetyt osoitteet viittauksissa parametreihin!). Taistelun päättymisen kriteeri asetetaan. Se voi liittyä kokonaisvahvuuteen tai kriittisen osion kulumiseen.

Usean osion mallissa on runsaasti parametreja, joiden reaalinen merkitys ei ole aina kovin selvä. Jos mallin avulla pyritään selvittämään parametrien vaikutusta taistelun lopputulokseen, erilaisia parametriyhdistelmiä on runsaasti. Tällainen tutkimus on toteutettava melko systemaattisesti, jotta sillä voisi katsoa olevan merkitystä.

Harjoituksia

18. Rakenna malli, jossa Sinisellä ja Keltaisella on kaksi osiota kummallakin. Tutki alkuvahvuuksien, vaikutuskertoimien ja osuuskertoimien vaikutusta taistelun lopputulokseen.

19. Rakenna malli, jossa Sinisellä ja Keltaisella on kaksi osiota kummallakin, ja toisen tuli on aluevaikutteista, toisen tähdättyä.

7.2 Monte Carlo -Lanchester

Lanchesterin mallit ovat deterministisiä. Taistelun lopputulos on alkuparametrien yksikäsitteinen funktio. Esitettyihin malleihin voidaan rakentaa sisään satunnaisuutta esimerkiksi asettamalla vaikutuskertoimet satunnaisluvuiksi. Kun taulukkoon rakennetaan, esimerkiksi edellä kuvatulla tavalla, taistelun lopputuloksen ilmaisin ja laskurit, voidaan taistelu ajaa useita kertoja.

Vaikutuskertoimien satunnaistamisessa on useita mahdollisuuksia. Kertoimet voidaan arpoa kerta kaikkiaan kutakin simulointikierrosta varten. Tällöin on järkevintä liikkua joissain haarukoissa $[a_0, a_1]$, ts. käyttää tyyppiä $a_0 + X(a_1 - a_0)$ tyyppistä rakennetta, missä X on @RAND-funktion tuottama välin $[0, 1]$ satunnaisluku. Taistelukentän kaoottisuutta saattaa paremmin mallintaa vaikutuskertoimen satunnaistaminen joka askeleella. Yksinkertaisimmin tämä tapahtuu korvaamalla edellä muodossa +A1−\$C\$1∗B1 ja +B1−\$D\$1∗A1 kirjoitetut kulumisaskeleen kaavat kaavoilla +A1−\$C\$1∗B1∗@RAND ja +B1−\$D\$1∗A1∗@RAND.

Harjoituksia

20. Rakenna edellä kuvatut toistomallit laskureineen ja tutki, miten erilaiset satunnaistamiset vaikuttavat simulaation lopputulokseen.

8 KIRJALLISUUTTA

[1] Lehtinen, Matti: Operaatioanalyysiä sotilaille. MpKK, Tekniikan laitos, Helsinki 2003.

[3] Lehtinen, Matti: Todennäköisyyslaskentaa sotilaille. MpKK, Tekniikan laitos, Helsinki 2002.

[2] Lehtinen, Matti: Tilastotiedettä sotilaille. MpKK, Tekniikan laitos, Helsinki 2002.

[4] Ross, Sheldon M.: A Course in Simulation. Macmillan Publishing Company, New York 1990.