Stockholms universitet

Modeller för reservoarer - köer och liknande fenomen

Robert Eriksson

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2011:2
Matematisk statistik
Juni 2011

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematisk statistik
Stockholms universitet
Kandidatuppsats **2011:2**,
http://www.math.su.se/matstat

Modeller för reservoarer - köer och liknande fenomen

Robert Eriksson*

Juni 2011

Sammanfattning

Många stokastiska fenomen kan beskrivas enligt följande mönster: Vi har en reservoar med ett inflöde, ett utflöde och en regel som garanterar att ingenting försvinner. Det typiska exemplet på detta är en kö, som byggs upp av ankommande kunder och avvecklas med att kunder betjänas. Det finns en omfattade teori för köer som bygger på olika antagande om ankomstprocessen och betjäningsprocessen. Andra snarlika exempel som tas upp är modeller för köpcentrum, trafikljus, dammar, datorkommunikation och försäkringsbolag. Dessa diskuteras och jämförs. En simulering för att testa trovärdigheten i en del av köteorin utförs också.

*Postadress: Matematisk statistik, Stockholms universitet, 106 91, Sverige. E-post: roberteriksson@hotmail.se. Handledare: Åke svensson.

Abstract

Many stochastic phenomena can be described as follows: We have a reservoir with an inflow, an outflow and a rule that ensures that nothing is lost.

The typical example is a queue, which is built up of arriving customers and settled with the customers served. There is a covering theory for queues based on different assumptions about the arrival process and service process. Other similar examples discussed and compared are models of shopping malls, traffic lights, ponds, computer communications and insurance companies.

A simulation to test the credibility of some of the queuing theory is performed as well.

Förord

Denna uppsats utgör ett examensarbete om 15 högskolepoäng och leder till en kandidatexamen i matematisk statistik vid Matematiska institutionen, Stockholms universitet.

Jag vill tacka min handledare, Åke Svensson, professor vid Stockholms universitet för vägledning, goda råd och värdefull inspiration under arbetes gång.

Innehåll

1 Introduktion

1.1 Syfte och frågeställning

Många stokastiska fenomen beskrivs genom en reservoir med ett inflöde, ett utflöde samt någon form av regel som gör att ingenting försvinner på vägen. Det typiska exemplet på detta är en regelrätt kö som byggs upp av ankommande kunder och avvecklas med att kunderna betjänas. Ur detta har det genom åren uppkommit en teori som bygger på olika antaganden. Syftet med denna rapport är att försöka beskriva ett antal typiska situationer där modellstrukturen kan beskrivas som ovan. Beroende på tillämpning kan de stokastiska fenomen som är värda att intressera sig för vara olika. Syftet är också att se likheter och olikheter mellan de olika situationerna och diskutera dessa. För att verkligen få grepp om en viss situation kommer någon simulering även göras med programmet Matlab.

Det jag vill få reda på genom detta kandidatarbete är i punktform

- Sätta in mig ordentligt i den grundläggande teorin.
- Vilka vanliga modeller byggs upp som beskrivningen ovan?
- Se likheten och skillnader mellan dessa.
- Lära mig att göra en regelrätt simulering.

2 Teori

2.1 Fördelningar

Dessa fördelningar kommer användas i rapporten.

2.1.1 Exponentialfördelnignen

En kontinuerlig slumpvariabel Y sägs vara exponentialfördelad med intensitet λ om täthetsfunktionen ges av

$$f_Y(y) = \lambda e^{-\lambda y},\ y \leq 0$$

En vanlig beteckning är $Y \sim \text{Exp}(\lambda)$

Väntevärdet $E(Y) = 1/\lambda$ och variansen $Var(Y) = 1/\lambda^2$

2.1.2 Poissonfördelningen

En disket slumpvariabel Y sägs vara Poissonfördelad med parameter λ om sannolikhetsfördelningen ges av

$$p_Y(k) = P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!},\ k = 0, 1, 2, 3...$$

En vanlig beteckning är $Y \sim \text{Po}(\lambda)$

Väntevärdet $E(Y) = \lambda$ och variansen $Var(Y) = \lambda$

2.1.3 Lognormalfördelningen

En kontinuerlig slumpvariabel Y sägs vara lognormalfördelad med parametrar μ och σ^2 om täthetsfunktionen ges av

$$f_Y(y) = \frac{1}{y\sigma\sqrt{2\pi}} e^{-\frac{(ln(y)-\mu)}{2\sigma^2}}$$

En vanlig beteckning är $ln(Y) \sim N(\mu, \sigma^2)$

Väntevärdet $E(Y) = e^{\mu+\sigma^2/2}$ och variansen $Var(Y) = e^{2\mu+\sigma^2}\left(e^{\sigma^2} - 1\right)$

2.2 Räkneprocess

En stokastisk process $N(t), t \geq 0$ sägs vara en räkneprocess om $N(t)$ representerar det totala antal händelser som inträffat vid tiden t. $N(t)$ måste alltså uppfylla följande
(1) $N(t) \geq 0$
(2) $N(t)$ är heltalsvärd
(3) Om $s < t$, så måste $N(s) \leq N(t)$
(4) För $s < t$ så anger $N(t) - N(s)$ antalet händelser i intervallet $[s, t]$

2.3 Poissonprocess

Poissonprocessen som fick sitt namn efter den franska matematikern Siméon-Denis Poisson (1781-1840) är en homogen process som känns igen genom en skalparameter λ, som även kallas intensiteten, så att antalet händelser i intervallet $(t, t+h)$ följer en Poissonfördelning med parameter λh. Man brukar definiera Poissonprocessen enligt.
(1) $N(0) = 0$
(2) Oberoende och stationära inkrement
(3) Antalet händelser i vilket intervall som helst av längd h är Poissonfördelad med parameter λh.

2.4 Kösystem som används

Man beskriver ofta kösystem enligt metoden $A/B/C$ där A innebär antaganden för ankomster, B innebär antaganden om betjäning och slutligen C är hur många kassor det finns eller som är öppna. De två kösystem som används i rapporten är $M/M/1$- och $M/G/3$-köer. Det första M:et antar Poissonfördelade ankomster, den andra M:et antar exponentiell avveckling och det andra G:et betyder en mer generell avvecklingsfördlening. Slutligen har den ena fallet en kassa öppen medan den andra har tre kassor öppna.

2.5 Markovkedjor

Låt X_n vara en stokastisk process som kan anta ett ändligt antal positiva värden. Om $X_n = i$ sägs processen vara i tillstånd i vid tiden n. Låt oss nu anta att när processen är i tillstånd i finns det en fixerad sannolikhet P_{ij} att den i nästa steg kommer vara i tillstånd j. Det kan skrivas som

$$P_{ij} = P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, ..., X_1 = i_1, X_0 = i_0)$$

för alla tillstånd $i_0, i_1, ..., i_{n-1}, i, j$ där $n \geq 0$. En sådan process är känd som en Markovkedja. Ekvationen ovan kan tolkas som att sannolikheten att hamna i ett visst framtida tillstånd X_{n+1}, givet alla de gamla tillstånden $X_0, ..., X_{n-1}$ och X_n, är oberoende av alla gamla tillstånd utom det senaste. En Markovkedja är alltså minneslös.

2.6 Förnyelseprocess

Om sekvensen av icke-negativa slumpvariabler $X_1, X_2, ..., X_n$ är oberoende och likafördelade där alla X_i är tiden mellan hopp så sägs räkneprocessen $N(t),\ t \geq 0$ vara en förnyelseprocess.

2.7 Födelse- och dödsprocess

Anta ett system vars tillstånd är representerat av antalet personer i systemet vid en viss tidpunkt. Anta vidare att när det är n personer i systemet tillkommer det nya ankomster med en exponentiell takt λ_n, samt att personer lämnar systemet med en exponeltiell takt μ_n. Alltså är, då vi har n personer i systemet, tiden till nästa ankomst exponentialfördelad med väntevärde $1/\lambda_n$ och oberoende av tiden till nästa person lämnar systemet som är exponentialfördelad med väntevärde $1/\mu_n$. Ett sådant system kallas födelse- och dödsprocess. Parametrarna $\{\lambda\}_{n=0}^{\infty}$ och $\{\mu\}_{n=0}\infty$ kallar respektive för (födelse)ankomsttakt och (döds)avgångstakt.

2.8 M/M/k förlustsystem

Ett förlustsystem i en $M/M/k$ kö innebär att kunder bara kan ankomma om minst en av k kassor är lediga. Om så inte är fallet är kunden förlorad ur systemet.

2.9 Lévyprocess

En Lévyprocess är en process $X(t)$, $t \geq 0$ som uppfyller följande villkor.
(1) Oberoende och stationära inkrement.
(2) $X(t)$ är kontinuerlig i sannolikhet, alltså för varje $\epsilon < 0$ gäller

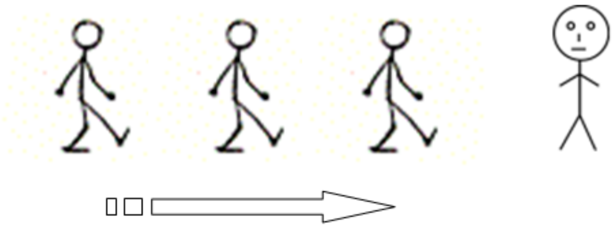
$$P(|X(t)| > \epsilon) \to 0 \text{ då } t \to 0^+$$

(3) Det existerar gränser både till höger $X(t+)$ och vänster $X(t-)$ och vi antar att $X(t)$ är "högerkontinuerlig", så att $X(t+) = X(t)$.

3 Urval av reservoirmodeller i olika situationer

3.1 Postkontor (M/M/1-kö)

Ett postkontor eller liknande serviceställe är en direkt tillämpning av köteorin ovan. Man intesserar sig för kötider, kölängder, antal personer som står i kö och så vidare. Kunder ankommer med en poissonprocess, detta medför att tiderna mellan deras ankomst blir exponentialfördelade med någon parameter. Kunderna tar därefter en nummerlapp och står i kö till det är deras tur i ett helt rättvist system, den som kommer först blir också betjänad först. När personen i kö väl kommer till kassan är betjäningstiden exponentialfördelad med en viss parameter.



Figur 1: Postkontoret är ett rättvist system där kunder ankommer, väntar på sin tur och blir betjänade.

3.2 Snabbköp (M/G/3-kö)

Ett snabbköp är en liknande men lite mer avancerad situation än postkontoret ovan. Detta går givetvis att modellera på en rad olika vis, detta är bara ett exempel.

Kunder anländer till snabbköpet[1] enligt en poissonprocess, tiderna mellan kunders ankomst blir således exponentialfördelade. Det första som händer är att kunden tar en kundvagn eller en korg. Vi antar att alla kunder använder sig av något av dessa. Kunden går sedan vidare in i affärens gångsystem. Varje snabbköp har $n = 1, 2, 3...$ antal gångar. Tiden det tar att hämta en vagn eller korg samt gå till den första gången är lognormalfördelad med väntevärde μ_{start} och varians σ^2_{start}. Observera att alla observationer är positiva. När kunden väl kommer till den första gången, vilket vi noterar gång #1 är sannolikheten att hon går nerför den gången p_1 och sannolikheten $1 - p_1$ att hon inte behöver något där och går vidare. Valet att gå ner i

[1]Buckley (2005): *Simulating Fuzzy Systems*. Springer Berlin / Heidelberg

de övriga gångarna är oberoende av valet att gå ner i gång #1, sannolikheterna är därför p_n att gå ner gång $\#n$ och $1-p_n$ att gå vidare för $n = 1, 2, 3...$

Då en kund bestämmer sig för att gå ner i en gång, låt oss säga gång $\#i$ bestäms tiden som spenderas där av en lognormalfördelning med väntevärde μ_i och varians σ_i^2. Antalet varar kunder lägger i sin korg eller vagn bestäms av en diskret sannolikhetsfördelnig, vilken vi väljer att kalla Sanno(i) där $1 \leq i \leq m$.

För enkelhetens skull antar vi att snabbköpet har 3 kassor öppna. Om antalet överstiger 3 är tillvägagångssättet likadant, men det blir lite svårare att förklara. Varje kassa har sin egen kö där, återigen för enkelhetens skull, kunderna snällt står och väntar till det är deras tur. Kunderna byter alltså inte kö då de väl ställt sig där. Låt K_i vara antalet personer i kö i, $i = 1, 2, 3$. Självklart väljer varje kund den kö som är kortast, detta kan uttryckas på följande sätt: (1) Om $K_1 \leq K_2$ är sann så väljer kunden kö $i = 3$ om $K_3 \leq K_1$ och kö $i = 1$ annars. (2) Om $K_1 \leq Q_2$ är falskt så väljer kunden kö $i = 3$ om $K_3 \leq K_2$ och kö $i = 2$ annars. Vi antar att varje kö i teorin har oändlig kapacitet.

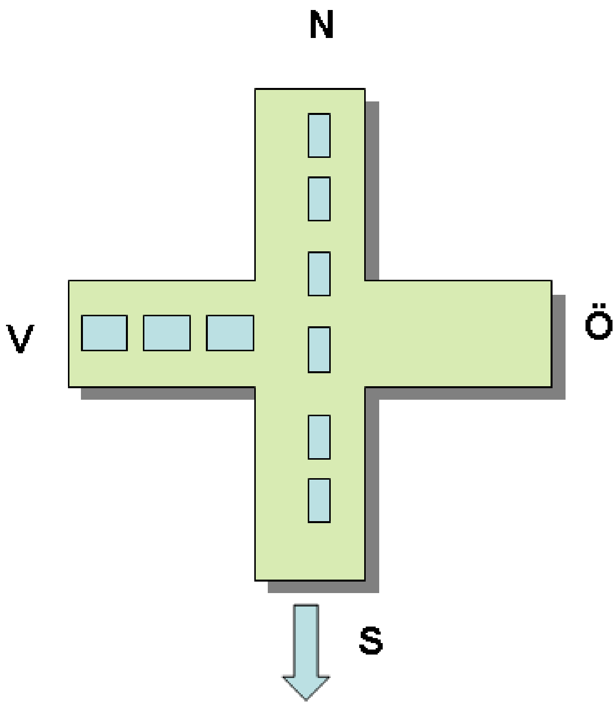
Om en kund måste vänta på sin tur i kassan tillkommer ett begrepp som ofta kallas impulsköp (många snabbköp lägger godis och tidningar samt extraprisvaror vid kassan). Antalet av dessa varor som kunden väljer att lägga vantarna på har ytterligare en diskret sannolikhetsfördelning som vi kallar Sanno(ik), ik för impulsköp. Till sist är tiden i kassan lognormalfördelad med väntevärde $M * \mu_{kassa}$ och varians σ_{kassa}^2. Där scannas alla varor samt betalning sker. M är antalet varor kunder har i sin korg eller i sin vagn och μ_{kassa} är den väntade tiden en kassör eller kassörska tar på sig att scanna en vara. Efter detta lämnar en förhoppningsvis nöjd kund snabbköpet.

3.3 Trafikkorsning

Vi tänker oss en korsning med två enkelriktade gator. Det kommer alltså bara bilar från ett håll på de respektive vägarna. Det ena hållet kallar vi södergående flödet och det andra östgående flödet. För att undvika en krock mellan bilarna finns ett trafikljus där trafiksignalen går i cykler grönt till rött och tillbaka till grönt och så vidare. Cyklen[2] börjar då trafikflödet från söder får grönt ljus och pågår åtminstone i g_1 sekunder. För varje bil som kör in i korsningen förlängs tiden för det gröna ljuset med f_1 sekunder. Dock är det maximalt grönt i g_2 sekunder. Då ljuset, vid någon tidpunkt, slår om till rött är det rött för båda sidor i en slumpmässig tidperiod. Detta ska simulera gult ljus samt extra uppstartningstid för den första bilen i östgående riktning.

[2]Lehoczky (1969): *Stochastic models in traffic flow theory: Intersection control.* Stanford University

Det är grönt ljus för det östra flödet i åtminstonde r_1 sekunder och för varje ny bil som kommer till korsningen förlängt den gröna tiden med f_2 sekunder dock maximalt till r_2 sekunder. Då trafikljudet slår om till rött är det rött för båda en viss tid precis som innan, efter detta börja cyklen om igen.



Figur 2: BIlar ankommer till systemet, vissa tvingar stanna för rött ljus medan andra kör rakt igenom.

3.4 Datakommunikation

Det finns många olika tillämpningar av datakommunikation. Här under följer några modeller för olika situationer

3.4.1 Ett multipellt databehandlingssystem

Vi tänker oss ett system som tar emot meddelanden och skickar vidare dessa. I systemet finns totalt N källor. Dessa källor är på eller av vid olika tidpunkter oberoende av varandra. En switch tar emot meddelanden med ett visst flöde beroende på hur många källor som är på och av. Samma switch skickar sedan vidare meddelanden. Vissa meddelanden går dock inte att skicka vidare direkt så dessa sparas i en lagringsenhet med oändlig kapacitet. Inga meddelanden försvinner alltså på vägen. Källorna agerar oberoende av varandra och tidpunkterna då dem är av eller på ses som oberoende exponentialfördelade slumpvariabler. Om nu $J(t)$ definieras som antalet på-källor vid tiden t leder detta till att $J = [J(t), t \geq 0]$ är en födelse-döds-process med tillstånd från 0 till N med födelseintensiteterna λ och dödsintensiteterna μ.

Låt nu $Z(t)$ vara buffertens innehåll vid tiden t. Modellens[3] tillstånd blir då

$$Z(t) = Z(0) + \int_0^t J(s)ds - \int_0^t r(Z(s), J(s))ds$$

där $r(x, j) = c$ om $x > 0$ och $r(x, j) = \min\,(j, c)$ om $x = 0$

Matematiskt ser detta tjusigt ut men vad det egentligen säger är att innehållet i bufferten vid tiden t är lika med vad som fanns i bufferten från början plus all ingående data, den data som ”föds in i bufferten”, under perioden $[0, t]$ minus all utgående data, som ”dör”, under samma period.

3.4.2 Datakommunikationsmodell som drivs med en M/M/1-kö

Vi får input av data från en $M/M/1$-kö. Så länge som systemet är upptaget ankommer data med ett konstant inflöde, låt oss kalla detta c_{in}. Dessa data skickas sedan vidare med ett maximalt utflöde c_{ut} som är mindre än c_{in}. Om nu $J(t)$ är kölängden som kan representera buffertinnehållet $Z(t)$ vid tiden t så fås modellen [4]

$$Z(t) = Z(0) + \int_0^t C_{0\{J(s)>0\}}ds - \int_0^t r(Z(s), J(s))ds$$

där $r(x, j) = 0$ då $x = j = 0$ och $r(x, j) = c_1$ annars.

3.5 Damm

Modellerandet för en damm beror i högsta grad på hur stor kapacitet dammen har. Här under följer några olika modeller beroende på hur dammen i fråga ser ut.

3.5.1 Modell för en damm med ändlig kapacitet

Definera X_{n+1} som mängden vatten i en lämplig enhet som har strömmat in i dammen under perioden $[n, n+1]$. Alla X_i för de intressanta i är oberoende likafördelade stokastiska variabler. Vi antar vidare att dammen har ändlig kapacitet, där maxnivån ges av c. På grund av detta kommer det på sikt bli

[3] Prabhu (1998): *Stochastic Storage Processes: Queues, Insurance Risk, Dams and Data Communication, 2nd edition.* Springer-Verlag New York, sid 166.

[4] a.a. sid 167.

en översvämning, och den faktiska tillförseln av vatten efter översvämningen är

$$\delta_{n+1} = \min(X_{n+1}, c - Z_n)$$

där Z_n är vattennivån i dammen vid tiden n. Det finns också en efterfrågan på vatten och mängden efterfrågat vatten vid n betecknar vi som ϵ_n där $\epsilon_1, \epsilon_2, ...$ antas vara oberoende likafördelade stokastiska variabler, samt att ϵ_n är oberoende av δ_n. Vatten frisläpps alltså på följande sätt

$$f(Z_n + \delta_{n+1}, \epsilon_{n+1}) = \min(Z_n + \delta_{n+1}, \epsilon_{n+1})$$

och vi får modellen[5].

$$Z_{n+1} = Z_n + \delta_{n+1} - f(Z_n + \delta_{n+1}, \epsilon_{n+1})$$

3.5.2 Modell för en damm med oändlig kapacitet

En damm med oändlig kapacitet kan givetvis inte råka ut för någon översvämning som ovan. Vi låter nu $X(t)$ beteckna mängden vatten som har strömmat in i dammen under en kontinuerlig tid på intervallet $[0, t]$. Vidare antas $X(t)$ vara en Lévyprocess utan drift. Frisläppningen av vatten kallar vi $r[Z(t)]$ där $Z(t)$ är mängden vatten i dammen vid tiden t. I detta fall är $r(x)$ en kontinuerlig ickeavtagande funktion för $x > 0$, samt givetvis $r(0) = 0$. Detta ger att mängden vatten vid tiden t ges[6]. av.

$$Z(t) = Z(0) + X(t) - \int_0^t r[Z(s)]ds$$

Alltså ett uttryck för $Z(t)$. Integralen i uttrycket representerar den totala frisläppningen av vatten under intervallen $[0, t]$.

Notera att om $r(x) = 1$ för $x > 0$ och $r(x) = 0$ för $x = 0$ är modellen analog med modellen för en ändlig damm som då får kapacitet $c = \infty$.

3.6 Försäkringsrisk

Försäkringsbolag måste se till att hålla reservkassa om en hög med större skador skulle komma. Bolagen gör omfattande stresstester för att ruin inte ska kunna ske med en större sannolikhet än 1 ruin på 200 år. Antalet anspråk på utbetalningar $X(t)$, även kallade fodringar som uppstår i tidsintervallet $[0, t]$ ges av den sammansatta Poissonfördelningen

$$\sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} P_n(x)$$

[5]Prabhu (1998): *Stochastic Storage Processes: Queues, Insurance Risk, Dams and Data Communication, 2nd edition.* Springer-Verlag New York, sid 4

[6]a.a. sid 5

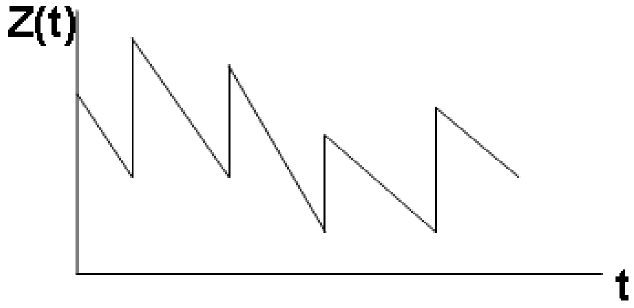
där $P_n(x)$ är fördelningen för antalet fodringar som verkligen betalas ut.

Bolaget erhåller premier från deras försäkringstagare med en konstant frekvens β. För att vara undvika ruin måste försäkringsbolaget hålla sig med en reservfond enligt ovan. Denna reserv ges vid tiden t av.

$$Z(t) = Z(0) + \int_0^t \beta\, du - X(t) = Z(0) + \beta t - X(t)$$

Här kan $Z(t)$ anta positiva eller negativa värden, men försäkringsbolaget är intresserade att hålla reservkassan stor nog för att undvika ekonomisk undergång över en ändlig eller oändlig tidsperiod. Bolagen vill beräkna en sannolikhet så att

$$P(Z(t) < 0) \leq 0.005$$



Figur 3: Beroende på antalet skador samt hur stora dessa skador är kommer reservfonden gå upp och ner med tiden. Även premieinkomsten kan vara olika om det kommer nya kunder eller kunder säger upp sin försäkring etc.

4 Simulering

Köer är ett fascinerande moment i vardagen. Detta skall nu illustreras i form av några simuleringar där det klart och tydligt utläses att ankomst- samt betjäningstiden spelar en stor roll.

Till och börja med ges en beskrivning av simuleringen, sedan presenteras resultaten i tabellform. Observera att fokus ligger på antal personer i systemet, eller mer korrekt - medelvärdet av dessa. Påpekas bör också att den negativa logaritmen av ett randomiserat slumptal är exponentialfördelat, simuleringen är alltså identisk med en $M/M/1$-kö med ankomstintensitet λ och betjäningsintensitet μ.

Först simuleras tiden då första kunden ankommer med negativa logaritmen av ett randomiserat slumtal delat med λ, dennes tid i kö är givetvis ingen alls. Betjäningstiden för första kunden simuleras med den negativa logaritmen av ett randomiserat slumptal delat med μ. Tiden då första kunden lämnar systemet är alltså ankomsttiden+betjäningstiden. Detta gäller givetvis inte för alla kunder där även väntetiden kommer spela in. Tiden då kund $i \geq 2$ lämnar systemet beskrivs som ankomsttiden+väntetiden+betjäningstiden. Beroende på valet av λ och μ kommer väntetiden och således antal personer i kö att ändras.

Om en kund måste vänta på sin tur eller ej beror såklart på om kassan är ledig eller ej, här finns bara en kassa till förfogande så det gäller att komma då systemet är rent för att undvika väntetid (kö). Om en kund har x personer före sig måste denne vänta på att alla de x ska bli betjänade innan det är hans/hennes tur. Det är alltså ett rättvist system.

Denna simulering går som sagt ut på att beräkna antal personer i systemet. En känd formel säger oss att denna skall vara

$$L = \frac{\lambda}{\mu - \lambda}$$

Jag kommer göra 5 simuleringar av total 10 000 ankommande personer och sedan ta medelvärdet för dessa. Det bör ge en rättvis bild. Detta kommer ske för varje val av λ och μ

λ	μ	Simulering	Teori
1/12	1/8	1.9909	2
2	3	2.008	2
3	4	2.934	3
5	6	4.306	5
1	2	0.99	1
2	4	1.03	1
7	8	5.478	7
3	6	0.9964	1
40	50	3.7304	4

Tabell 1: Skillnader mellan det teoretiska värdet på L och simuleringsvärdet. Dessa skiljer sig olika mycket beroende på valet av λ och μ.

Vi ser nu tydligt att simuleringen stämmer bra till teorin på låga antal, $L = 1, 2, 3$. Men ju högre L vi förväntar oss desto mer diffar simuleringen. Detta kan delvis bero på hur jag definierar ankomst- och betjäningsprocessen. Valet att $\mu > \lambda$ är medvetet, annat fall skulle L blir negativt. Ett negativt medeltalet av kunder i systemet är givetvis samma sam som att medeltalet är noll och därmed inte så intressant.

Ankomsten simuleras med

$$A = \frac{-log(\text{rand})}{\lambda} + \text{föregående ankomsttid.}$$

och betjäningstiden simuleras med

$$B = \frac{-log(\text{rand})}{\mu}$$

Då kvoten λ/μ ligger nära 1 verkar simuleringen ha en tendens att inte passa teorin särskilt bra. Tittar vi på formeln $L = \lambda/(\mu - \lambda)$ är det inte så konstigt då nämnaren i så fall ligger nära noll. För fortsatt utveckling av detta bör man alltså välja λ och μ så att kvoten inte är nära 1.

5 Likheter och Olikheter

Alla stokastiska modeller ovan har ett inflöde, ett utflöde samt en regel att inget försvinner på vägen. Det är den stora likheten. Skillnader däremot kan förekomma beroende på vilket fenomen vi pratar om. Detta kapitel diskuterar just detta.

I det enklaste fallet postkontor som följer direkt av den vanliga köteorin är man normalt intresserad av saker som genomsnittlig kölängd, antal personer i kö, hur långt tid det normalt tar att komma fram till betjäning o.s.v. Det finns formler och teori för detta. Men den gemensamma faktorn för kunden är att han/hon kommer in, gör sitt ärende och går ur. Det händer inget speciellt på vägen. Om vi går vidare till ett snabbköp eller stormarknad blir det genast lite mer komplicerat även om det bygger på samma grunder. Här kan man göra det hur enkelt som helst eller hur svårt som helst beroende på ens preferenser, ju mer verklighetstroget desto svårare blir det. Jag har valt att lägga mig någonstans mitt emellan. De vanliga antaganden gäller, dvs att kunder ankommer med en Poissonprocess och blir betjänade med exponentialfördelningar. Det intressanta är dock vad som händer där emellan. Mitt exempel utgår starkt från lognormalfördelningen hur många varor kunden tar samt en viss sannolikhet att kunden går till en viss gång. Här tänker vi oss ett slags korridorssystem där gångarna med varor ligger parallellt med varandra. När sedan kunderna spatserar på jakt efter varor är det alltså en viss sannolikhet att han/hon går in i gång 1, respektive gång 2 o.s.v. Detta kan anses mer eller mindre realistiskt. Svagheten i modellen är att olika kunder vill ha olika varor så normalfördelningen att alla ska vara lika faller. Det är dock en modell som är hyfsat lätt att simulera och med ett stort antal kunder bör skillnaderna på lång sikt jämna ut sig någorlunda.

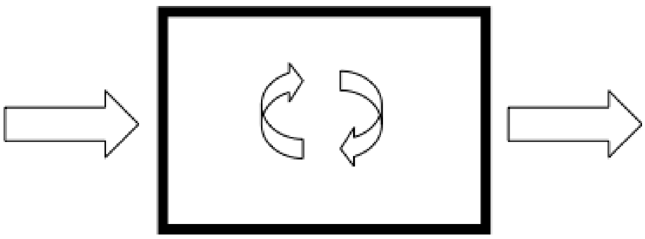
I avsnittet trafikkorsning ankommer bilar med en viss frekvens, beroende på denna frekvens blir köerna vid rött ljus långa eller korta. Och därav beror tiden då det är grönt ljus också av samma frekvens ty det gröna ljust förlängs med varje bil som kör. Dock maximalt till en viss tid. Skillnaden här emot de ovanstående fenomenen är att stokastiken går mer i cirklar. Grönt ljust, rött ljus för att sedan börja om igen med samma förutsättningar.

Datakommunikationen har jag valt att beskriva med två olika metoder, den första med en födelse- och dödsprocess och en andra med ett $M/M/1$-kösystem.

De stokastiska modellerna för en damm är beroende på om dammen är ändlig eller oändlig. En oändlig damm finns givetvis inte i praktiken, men om den är tillräckligt stor eller om efterfrågan på vatten är tillräckligt hög kan man utgå från att den är just oändlig då den aldrig blir full och således inte svämmar över. En skillnaden mellan de både är att vattnet kan frisläppas på två sätt då dammen är ändlig, nämligen att den är full och svämmar över eller att någon efterfrågar vatten. I en oändlig damm däremot kan vatten bara frisläppas vid efterfrågan. Vi ser också lite skillnader i modeller, där inströmningen av vatten ses i det ändliga fallet bara som oberoende likafördelade variabler, medan i det oändliga fallet ses inströmningen som en Lévyprocess.

Ett försäkringsbolag måste ha vissa matematiska regler för att inte gå omkull. Enligt krav ska sannolikheten att ett bolag går i konkurs vara mindre eller lika med en halv procent. Detta för att skydda kunderna. Speciellt inom livförsäkring är detta enormt riktigt. Men också inom sakförsäkring. För att undvika detta måste försäkringsmatematiker (aktuarier) inte bara räkna på risken för olyckor utan även avsätta kapital, så kallade reserver för varje försäkringsfall.

Det intressanta med alla dessa olikheter är att det också finns en stor likhet. Oavsett om det gäller en vanlig kö, en damm, datakommunikation, försäkringsbolag eller andra liknande fenomen som inte tas upp här så ser systemet likadant ut. Från början finns ett visst innehåll. Detta fylls sedan på, något händer, för att slutligen avta. Se figur 1 för en simpel illustration.

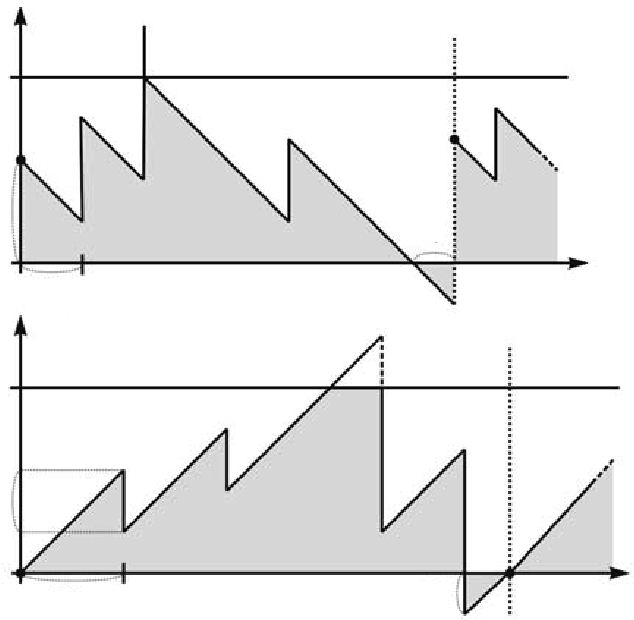


Figur 4: Ett inflöde till en reservoir där någonting hände beroende på tillämpningsområde och ett utflöde.

Intressant är också att vid simulering av olika områden är koden mycket lik varandra. Ett exempel[7] visas i figor 6 där den övre grafen är väntetid i en M/G/1-kö upp till en viss nivå och den undre grafen är riskreserven. Då reserven går över stecket säger vi att reservkassan är full. Viktigt här är att y-axeln på den övre grafen motsvarar tid och x-axeln motsvarar nya kunder. I den undre grafen är det omvänt. T.ex. motsvarar tiden då inga kunder kommer i den övre grafen underskottet i reserven i den undre grafen.

[7] Löpker & Perry (2010): *The idle period of the finite G/M/1 queue with an interpretation in risk theory.* Queueing Syst 64: 395-407

Påpekar bör också att precis samma data som simulerats i bägge fallen, det enda som skiljer är intresseområde.



Figur 5: Beroende på tillämpningsområde kan modellen med få justeringar se likadan ut men man tittar på andra saker.

6 Diskussion och slutsats

Alla exemplen ovan är så kallade lagringsprocesser (eng: storage processes) och beroende på tillämpning är dessa användbara i en rad olika situationer. Allt från en regelrätt kö till ett försäkrings- eller finansbolag. En av fördelarna med denna form av process är att den är relativt lätt att göra en vettig simulering på och dessutom kan samma simulering ofta användas till andra tillämpningsområden, se kapitel 5.

Många kurser bygger starkt på köteori och tanken har väl slagit mig vad man egentligen ska ha så mycket köteori till. Svaret är uppenbarligen att teorin kan utvecklas långt utanför köer till verkligt intressant och samhällsnyttiga områden. Jag känner att jag lärt mig ett nytt tankesätt, jag ser annorlunda på de stokastiska processerna nu än innan.

7 Referenslista

Alm& Britton (2008): *Stokastik.* Liber

Buckley (2005): *Simulating Fuzzy Systems.* Springer Berlin / Heidelberg

Johansson (2008): *Matematiska modeller inom sakförsäkring.* Kompendium, Stockholms Universitet

Lehoczky (1969): *Stochastic models in traffic flow theory: Intersection control.* Stanford University

Löpker & Perry (2010): *The idle period of the finite G/M/1 queue with an interpretation in risk theory.* Queueing Syst 64: 395-407

Prabhu (1998): *Stochastic Storage Processes: Queues, Insurance Risk, Dams and Data Communication, 2nd edition.* Springer-Verlag New York

Ross (2006): *Introduction to probability models, 9th edition.* Academic Press

A Appendix

A.1 Matlab-kod till simulering

```
lambda=x;
mu=y;
nollmatris=zeros(20,5);

%Antal kunder k-20
k=10020;

%Klockslag för kundens ankomst
ankomst=-log(rand)/lambda;

%Kundens väntetid i kö
wait=0;

%Kundens betjäningstid service=-log(rand)/mu;

%Klockslag då kunden lämnar systemet avgang=ankomst+wait+service;

%Antal personer i kö vantar=0;

lastline=[ankomst wait service avgang vantar];
data=[nollmatris;lastline];
for i=21:k
intervall=-log(rand)/lambda;
ankomst=lastline(1)+intervall;
wait=max((lastline(2)+lastline(3)-(ankomst-lastline(1))),0);
service=-log(rand)/mu;
avgang=ankomst+wait+service;


if ankomst¡data(i,4) && ankomst>data(i-1,4)
vantar=1;
elseif ankomst¡data(i-1,4) && ankomst>data(i-2,4)
vantar=2;
elseif ankomst¡data(i-2,4) && ankomst>data(i-3,4)
vantar=3;
elseif ankomst¡data(i-3,4) && ankomst>data(i-4,4)
vantar=4;
elseif ankomst¡data(i-4,4) && ankomst>data(i-5,4)
vantar=5;
elseif ankomst¡data(i-5,4) && ankomst>data(i-6,4)
```

```
vantar=6;
elseif ankomst¡data(i-6,4) && ankomst>data(i-7,4)
vantar=7;
elseif ankomst¡data(i-7,4) && ankomst>data(i-8,4)
vantar=8;
elseif ankomst¡data(i-8,4) && ankomst>data(i-9,4)
vantar=9;
elseif ankomst¡data(i-9,4) && ankomst>data(i-10,4)
vantar=10;
elseif ankomst¡data(i-10,4) && ankomst>data(i-11,4)
vantar=11;
elseif ankomst¡data(i-11,4) && ankomst>data(i-12,4)
vantar=12;
elseif ankomst¡data(i-12,4) && ankomst>data(i-13,4)
vantar=13;
elseif ankomst¡data(i-13,4) && ankomst>data(i-14,4)
vantar=14;
elseif ankomst¡data(i-14,4) && ankomst>data(i-15,4)
vantar=15;
elseif ankomst¡data(i-15,4) && ankomst>data(i-16,4)
vantar=16;
elseif ankomst¡data(i-16,4) && ankomst>data(i-17,4)
vantar=17;
elseif ankomst¡data(i-17,4) && ankomst>data(i-18,4)
vantar=18;
elseif ankomst¡data(i-18,4) && ankomst>data(i-19,4)
vantar=19;
elseif ankomst¡data(i-19,4) && ankomst>data(i-20,4)
vantar=20;
else vantar=0;
end

lastline=[ankomst wait service avgang vantar];
data=[data; lastline];
end

data=data(21:k,:);

bar(data(:,1),data(:,5))

mean(data(:,5))
```