

Simulering av ett multi-skill callcenter

Med varierande genomsnittlig betjäningstid
beroende på agenters kunskapsnivå

FILIP GAUFFIN



**KTH Datavetenskap
och kommunikation**

Simulering av ett multi-skill callcenter

Med varierande genomsnittlig betjäningstid
beroende på agents kunskapsnivå

F I L I P G A U F F I N

DD143X, Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik 300 högskolepoäng
Kungliga Tekniska Högskolan år 2012
Handledare på CSC var Johan Boye
Examinator var Mårten Björkman

URL: [www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2012/
gauffin_filip_K12029.pdf](http://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2012/gauffin_filip_K12029.pdf)

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Sammanfattning

Den här rapporten undersöker effekten av att använda varierande genomsnittlig betjäningstid beroende på agentens kunskapsnivå i ett *multi-skill callcenter*. Givet en modell för att skicka samtal till en agent simuleras callcentret för olika antal kompetenser, tryck och genomsnittlig betjäningstid. Syftet var att avgöra effekten och om fler kompetenser per agent alltid betyder bättre effektivitet som det gör hos ett multi-skill callcenter utan varierande genomsnittlig betjäningstid. Simuleringarna av callcentret visade att till en början blev effektiviteten bättre när agenterna fick fler kompetenser och sedan blev den sämre. Bästa resultat gav simuleringarna vid användning av två kompetenser per agent för alla scenarier som använde varierande genomsnittlig betjäningstid.

Abstract

The purpose of this report was to find the effect of using varying average service-times depending on the agents skill for different types of calls in a multi-skill call center. Given a model for distributing calls to agents a call center is simulated for different average service-times. The purpose was to determine the effect of introducing varying service-times and if introducing multiple skills always leads to better performance. The simulations showed that when introducing more skills than one, the result became better. However it also showed that there is a clear boundary when the effectiveness started to get worse by introducing yet another skill. Best result was achieved by using two skills for almost all simulations where the exceptions were using three skills.

Innehåll

1	Introduktion	2
2	Problemformulering	2
3	Bakgrund	3
3.1	Termer och begrepp	3
3.2	Modellering av ett multi-skill callcenter	3
3.3	Relaterad litteratur	4
3.3.1	SBR-Modell	4
3.4	Matematik	4
3.5	Simuleringsmodell	5
4	Metod	5
4.1	Simulering	5
4.2	Utdata	6
4.3	Specifikation av callcenter som simuleras	6
4.4	Ankomster och betjäning	7
4.4.1	Varierande genomsnittlig betjäningstid	7
4.5	Verktyg	7
4.6	Antaganden och begränsningar	7
5	Resultat	8
5.1	Simulering utan skillnad i genomsnittlig betjäningstid	8
5.1.1	Matematisk beräkning av ett callcenter	9
5.2	Simulering med liten skillnad av genomsnittlig betjäningstid	10
5.3	Simulering med stor skillnad av genomsnittlig betjäningstid	11
5.4	Simulering med en primär kompetens och övriga sekundära	13
6	Slutsats	14
A	Invertering av exponentialfunktionen	18

1 Introduktion

Då ett callcenter hanterar samtal av olika typer med agenter som har olika kompetenser, eller skills, kallas det för ett *multi-skill callcenter*. När företag växer eller slås ihop kan det vara både nödvändigt och fördelaktigt att lära agenterna olika kompetenser. Om t ex två callcenter slås ihop men ingen träning görs för att få agenter att ta hand om båda typerna av samtal har man i princip två separata callcenter identiska med hur de såg ut innan ihopslagningen. Ger man nödvändig träning finns det fler agenter som kan ta ett givet samtal och det är större chans att samtal inte behöver vänta. Det kan leda till att färre agenter behövs och kostnader minskar. Problemet som dyker upp i ett multi-skill callcenter är hur man ska hantera vem som ska ta vilket samtal. I det optimala fallet där alla agenter har alla kompetenser är det enkelt eftersom callcentret agerar som om det bara finns en kompetens, men om så inte är fallet behövs ett system som effektivt skickar samtal till en lämplig agent. Det man vill undvika är att agenter är lediga medan det finns väntande samtal för att alla agenter som kan ta hand om samtalen redan är upptagna. Givet en mängd samtal är det möjligt att skapa en algoritm som fördelar samtalen till rätt agent på effektivaste sätt, men eftersom det inte är känt på förhand vilka typer av samtal som kommer in vid vilken tidpunkt behövs en modell som fördelar samtalen i den mån de kommer och som hanterar det med avseende på callcentrets effektivitet. Notera att det i praktiken inte är genomförbart att göra om ett multi-skill callcenter med många kompetenser genom att träna agenterna så att det beter sig som om det bara finns en kompetens.

2 Problemformulering

Anta att vi har ett callcenter där kunskapsnivån hos agenterna varierar. T ex kan kunskapsnivåerna för en civilingenjör i datateknik delas upp på följande vis: Bäst kunskap inom data, bra kunskaper inom matematik, mindre bra kunskaper inom fysik, vilket stämmer relativt sett om man också har matematiker och fysiker i callcentret. Problemet är att dessa olika kunskapsnivåer kan påverka verksamheten och analysen av ett callcenter om ingenjörerna antas ta hand om de tre typerna lika effektivt. Ta till exempel ett callcenter som hanterar språk. En agent skulle kunna ta emot samtal som gäller dennes modersmål såväl som andra inlärd språk. När ärenden som gäller modersmål kommer går det snabbare än för andra språk. Att applicera en modell för multi-skill callcenter utan att ta hänsyn till detta kan få negativa konsekvenser. Antingen genom att man antar att den genomsnittliga betjäningstiden är kortare än vad den egentligen är då man tittar på de agenter som är effektivast, eller att den blir mycket högre för att man skickar samtal till agenter med längre betjäningstid när det finns lämpligare agenter lediga. Detta kan

leda till att fler agenter än vad som är nödvändigt behövs.

För att ta reda på hur varierande genomsnittlig betjäningstid påverkar effektiviteten har jag simulerat callcenter med en given modell för att tilldela samtal till agenter och olika antal kompetenser per agent med olika skillnader i genomsnittlig betjäningstid mellan olika kunskapsnivåer. En av de sakerna jag tittat på är om ett callcenter tjänar på att utbilda agenter för att kunna ta hand om så många samtalstyper som möjligt om de genomsnittliga betjäningstiderna varierar beroende på kunskapsnivå.

3 Bakgrund

3.1 Termer och begrepp

Här beskrivs de termer som läsaren bör känna till för att förstå resten av texten. Jag har valt att använda de engelska termerna eftersom de är vanligt förekommande i engelska texter inom ämnet.

ASA – Average Speed of Answer – Genomsnittlig väntetid för ett samtal i kö

AWT – Acceptable Waiting Time – Acceptabel väntetid för ett samtal (bestäms som mål)

SBR – Skilled-based Routing – Samtalsfördelning baserad på agenters kompetenser i ett multi-skill callcenter

SL – Service Level – Servicenivå. Står för andelen samtal som fått vänta längre än AWT, alternativt andelen samtal som inte behövt vänta längre än AWT. Litteraturen går isär inom detta.

FIFO - First-In First-Out - Beskriver systemet för en kö: Först in, först ut.

3.2 Modellering av ett multi-skill callcenter

Ett multi-skill callcenter kan modelleras som en matris där posterna i matrisen står för kompetenser. Varje rad i matrisen beskriver vilka kompetenser en specifik agent har. Dessutom beskriver den också kunskapsnivån för varje kompetens som varje agent har. Till exempel har agent 1 i matris M kompetens 1 som primär kompetens och kompetens 3 som sekundär, men saknar en tredje kompetens.

$$M = \begin{pmatrix} 1 & 3 & - \\ 2 & 1 & 3 \\ 3 & 2 & - \\ 2 & 3 & 1 \\ 1 & - & 2 \end{pmatrix}$$

3.3 Relaterad litteratur

För den bakomliggande matematiken och sätt att teoretiskt mäta effektiviteten för callcenter har Koole, G skrivit *Call center mathematics* [Koole, 2007] som handlar om hur man teoretiskt mäter effektiviteten hos ett callcenter med en kompetens med hjälp av Erlang-C- och Erlang-X-formlerna.

Wallace, R och Whitt, W. har skrivit rapporten *A staffing algorithm for call centers with skill-based routing* [Wallace och Whitt, 2005] där de definierar ett multi-skill callcenter och presenterar en modell för SBR och gör simuleringar med modellen på callcentret för att mäta dess effektivitet. Det de tittar på är effektiviteten hos det ursprungliga callcentret som de definierat med olika ankomstintensiteter och testar vad som händer för olika antal kompetenser per agent. Dessutom testar de vad som händer om antalet agenter och som köplatser förändras. Slutsatsen är att även med begränsad antal kompetenser per agent kan man uppnå ett resultat som likna det när alla agenter har alla kompetenser.

3.3.1 SBR-Modell

Modellen för SBR skapad av [Wallace och Whitt, 2005] tar mindre hänsyn till SBR och mer hänsyn vilka kompetenser agenterna har. D v s med denna policy är det viktigare för ett företag att se till att agenterna har rätt kompetenser än att använda ett system som passar de agenter som företaget redan har. En rimlig modell eftersom om man måste anpassa sig till tillgängliga agenter kan det innebära dyra specialskräddade lösningar för SBR och införskaffandet av nya lösningar när callcentrets situation förändras.

I modellen har varje agent kompetenser med olika prioritet i enlighet med sin kunskapsnivå som matris M i sektion 3.2. Tex kan en agent ta samtal av typ 1 och 3 där typ 1 är den primära kompetensen och typ 3 den sekundära. Samtal som kommer in skickas först till agenter som har samtalet som primär kompetens, sen till agenter som har det som sekundär, osv. Om det finns flera agenter som kan ta samtalet med sin primära kompetens skickas samtalet till den agent som har varit ledig under längst tid. Modellen som presenteras tar dock inte hänsyn till om det finns någon skillnad i genomsnittlig tid för betjäningen beroende på kunskapsnivå det vill säga om det tar längre tid för agenten ovan att ta hand om samtal av typ 3 än det tar för samtal av typ 1.

3.4 Matematik

Här kommer en kort beskrivning av den matematik som är relevant för den här texten. För att fullständigt förstå detta bör läsaren ha viss kunskap om markovprocesser och köteori [Engel och Grandell, 2006].

Ett callcenter kan modelleras som ett Markovskt kö-system $M/M/C/K$, som i [Wallace och Whitt, 2005] eller den mer generaliserade modellen $M/GI/C/K+GI$ i [Whitt, 2005] där man även tar hänsyn till kunder som

lämnar systemet efter att ha väntat för länge. Med hjälp av ankomstintensiteten λ och den genomsnittliga betjäningstiden μ fås trycket $\lambda * \mu = a$ mot callcentret och intensiteten blir a/C . Om a/C är större än 1 betyder det att det i genomsnitt ankommer fler samtal än vad agenterna kan hantera vilket man ska försöka undvika eftersom kön då växer konstant tills att nästan alla samtal blockeras. I fallet med ett single-skill callcenter kan man, givet AWT och trycket a , använda Erlang-C-formeln för att beräkna ASA och SL dessutom kan man använda Erlang-X-formeln för att beräkna ett mer avancerat callcenter med bara en kompetens där även blockerade samtal, antal telefonlinjer, kunders tålamod och andel kunder som lämnar systemet efter att ha fått vänta för länge finns med [Koole, 2007]. I fallet med ett multi-skill callcenter är det inte praktiskt genomförbart därför används ofta Monte-Carlo simuleringar eller diskret-händelse simuleringar.

3.5 Simuleringsmodell

Man kan simulera ett kontinuerligt system som t ex ankomsterna av kunder specificerad av en poisson-process genom dela upp simuleringen i ett antal tidsenheter. Vid varje tidsenhet kan det ske händelser, till exempel kan det ankomma en kund, två kunder osv. Utanför tidsenheten händer ingenting. Detta kallas för en diskret-händelse simulering och används av [Anton, Babat och Paul 1999] för att förbättra SL i ett callcenter. Vid varje tidsenhet loggar man det som händer för att senare sammanställa resultatet.

Då tiden mellan två händelser i en poisson-process är exponentialfördelad (precis som betjäningstiden) ger inversen av exponentialfördelningen en specifik tidsperiod X för en given sannolikhet U och ett givet väntevärde λ . Eftersom sannolikheten U (0-100) är likafördelad går det att använda en relativt enkel slumpgenerator för att få slumpmässiga tidsintervall X som sedan kan användas i simuleringen [Winter, 2004]. Betjäningstiderna fås med samma metod. Se appendix A för mer om invertering av exponentialfunktionen.

$$X = -\lambda * \ln(U) \tag{1}$$

4 Metod

4.1 Simulering

För simuleringen kommer modellen för SBR av [Wallace och Whitt, 2005] användas. Modellen fungerar bra för det här ändamålet eftersom det är en deterministisk modell som skickar ett samtal till den agent av de som är lediga som hör högst kunskapsnivå och lägst genomsnittlig betjäningstid för samtalstypen. Två olika kunskapsscenarier simuleras, en där skillnaden i genomsnittlig betjäningstid är liten per kunskapsnivå och en där skillnaden är

stor.

Simuleringen av ankomster och betjäningstid görs med hjälp av formel (1) där λ antar antingen det genomsnittliga värdet för tiden mellan ankomster eller den genomsnittliga betjäningstiden. För betjäningen tolkas X som tiden för en betjäning och för ankomsterna räknas X som tiden mellan två ankomster. Värdet U antar slumpmässigt värden mellan 0 och 1. För varje uppsättning kompetenser och tryck körs en simulering 10 gånger och medianen används för utdatan. Varje simulering kör i 1000000 abstrakta minuter (ca 2 år) och antal samtal under den tiden blir ca 7740000, 8400000 eller 9000000 samtal beroende på vilken ankomstintensitet som används.

4.2 Utdata

För att verifiera effektiviteten på ett callcenter kommer i huvudsak tre parametrar användas. Andel blockerade samtal, ASA och SL. För ASA och SL räknas inte de blockerade samtalen med utan endast de samtal som fått sitt samtal besvarat räknas med. För SL kan man använda två gränser, till exempel efter 30 sekunder väntetid har man fått dålig service och efter 2 minuter väntetid har man fått ännu sämre service. Här kommer bara den första gränsen, 30 sekunder, att användas.

Utöver det kommer också användningen att visas. Det vill säga hur stor andel av tiden agenterna aktivt arbetar med ett samtal. Det finns med bland annat för att kunna avgöra om det finns för få eller för många agenter men framför allt för att det är enkelt att skapa ett callcenter som uppfyller alla krav om man inte bryr sig om antalet agenter och de lönekostnader som de medför.

4.3 Specifikation av callcenter som simuleras

De callcenter som simuleras kommer likna ett som användes av [Wallace och Whitt, 2005](#) med avseende på antal agenter, max antal kompetenser och antal telefonlinjer. Till skillnad från [Wallace och Whitt, 2005](#) kommer dessa variabler inte att varieras under eller mellan simuleringarna.

Callcentret har 90 agenter där antal kompetenser per agent varierar från en kompetens till sex kompetenser mellan simuleringarna. Kompetenserna är fördelade så att grupperna med agenter som har en viss kompetens för en viss kunskapsnivå är lika stora. Eftersom det finns 6 kompetenser och 90 agenter har dessa grupper alltid 15 agenter. Totalt finns det 120 telefonlinjer, en för varje agent och 30 för köande samtal. Kön fungerar som en FIFO-kö. Alternativt skulle man kunna använda ett kösystem där man försöker ta samtal som ligger under gränsen för SL för att förbättra statistiken medan samtal som redan fått vänta längre än gränsen för SL får vänta ännu längre. Samtal som inte kan tas emot av en agent och som inte får plats i kön blockeras och får lämna callcentret utan att bli betjänade.

4.4 Ankomster och betjäning

När samtal ankommer styrs av en poissonprocess som använder ankomstintensiteten 7,74 samtal per minut, 8,4 samtal per minut eller 9 samtal per minut. För en betjäningstid på 10 minuter fås trycket 0,86 0,93 respektive 1,0 som fortsättningsvis kommer refereras till som lågt tryck, normalt tryck och högt tryck. Minns att ett tryck över 1 inte är rimligt för ett callcenter, se sektion [3.4](#). Den genomsnittliga betjäningstiden kommer variera beroende på kunskapsnivå men kommer utgå från 10 minuter. När varierande genomsnittlig betjäningstid används kommer agenter ha genomsnittliga betjäningstid 10 minuter för sin primära kompetens men för andra kompetenser kommer tiden vara högre.

4.4.1 Varierande genomsnittlig betjäningstid

För varierande genomsnittlig betjäningstid kommer skillnaderna som används vara 5% som liten skillnad i genomsnittlig betjäningstid och 15% som stor skillnad skillnad i genomsnittlig betjäningstid. Eftersom den genomsnittliga betjäningstiden utgår från 10 minuter innebär det skillnader på 0.5 minuter respektive 1.5 minut mellan kompetensnivåerna. Om skillnaden i procent betecknas som X blir den genomsnittliga betjäningstiden för varje kompetensnivå n :

$$10 + 10 * X(n - 1) \tag{2}$$

4.5 Verktyg

För att implementera simuleringen används programmeringsspråket Java och särskilt klassen Random används för att generera slumpantal för variabeln U i formel (1) för ankomster och betjäningstider. Dessutom används Erlang-X-kalkylatorn [Kooole](#) för att matematiskt räkna fram de variabler som simuleringen får ut för ett callcenter med en kompetens för att jämföra med de simulerade värdena.

4.6 Antaganden och begränsningar

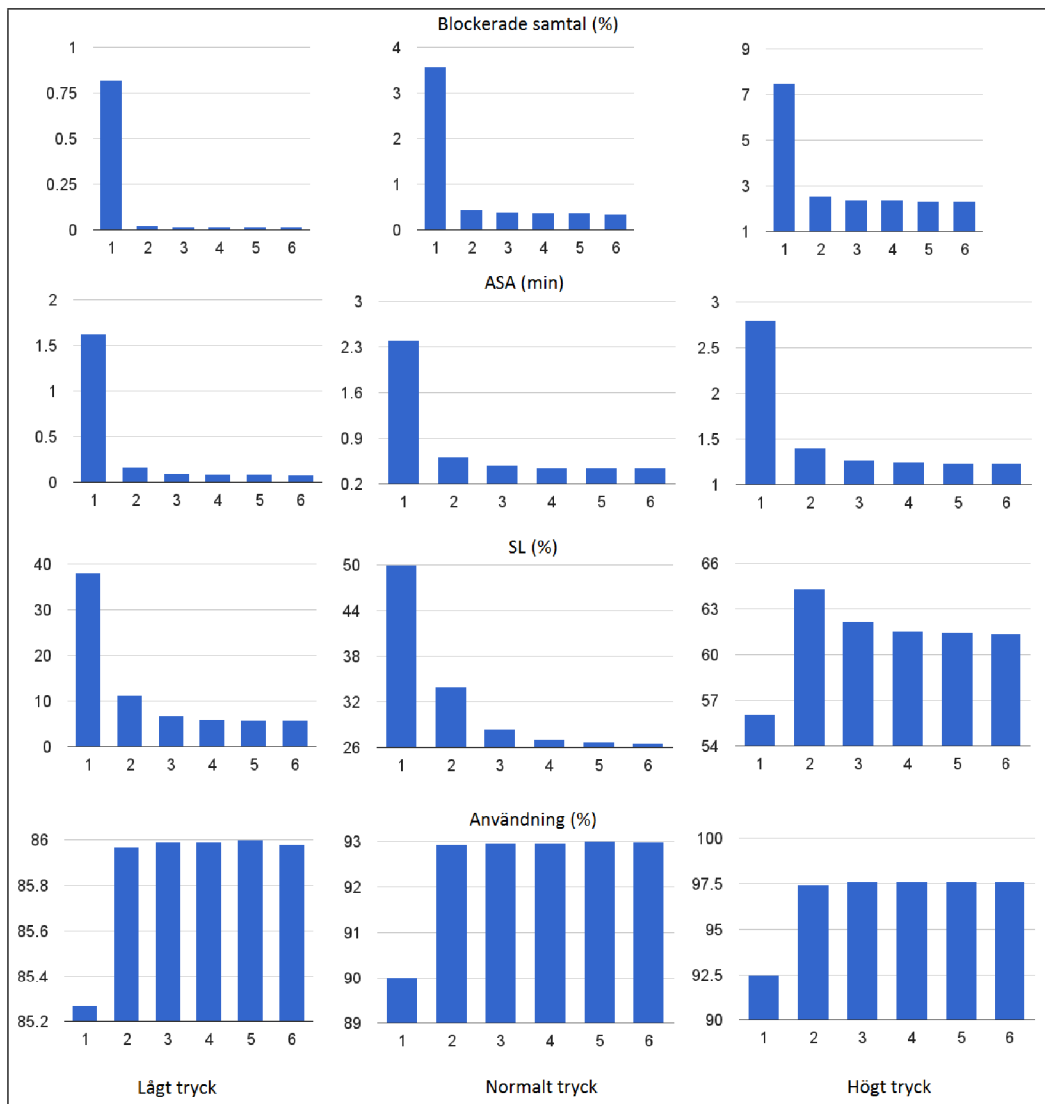
Ingen hänsyn tas till att det kan ta olika lång tid att matcha ett samtal med en ledig agent. Jag antar att det tar lika lång tid för alla samtal av alla typer och tar inte särskild hänsyn till det i simuleringen. Återuppringning för blockerade samtal finns inte med och i scenarier med hög blockeringsandel kommer resultatet inte återspegla ett riktigt callcenter om man räknar med att en viss del kunder ringer tillbaka efter att de blockerats. När varierande genomsnittlig betjäningstid införs kommer trycken 0,86 0,93 och 1,0 inte längre att gälla eftersom betjäningstiden varierar.

5 Resultat

5.1 Simulering utan skillnad i genomsnittlig betjäningstid

För att jämföra resultatet från införande av varierande genomsnittliga betjäningstider används ett callcenter med fixerad genomsnittlig betjäningstid på 10 minuter. Resultatet för andel blockerade samtal, ASA och SL visas i figur 1 såväl som användningen visas. Som man kan se ligger den största skillnaden i andel blockerade samtal, ASA och SL när man använder två kompetenser istället för en. Även mellan två och tre kompetenser finns det märkbara förbättringar men för fler kompetenser än tre är det relativt små förbättringar. Ett intressant resultat är att SL skiljer sig från resultatet i övrigt när trycket är högt vilket man kan se i diagrammet över SL i figurer där SL blir sämre när man använder två kompetenser istället för en vid högt tryck. Att det är högt tryck medför att samtal ofta behöver ställa sig i kö men eftersom ingen agent kan ta hand om mer än en typ av samtal kan det uppstå en situation där kön är full trots att agenter är lediga. När ett samtal kommer som lediga agenter kan ta går det samtalet före i kön och bidrar till bättre SL. Detta kan uppstå med 2-5 kompetenser också men det visar sig inte påverka lika mycket som för en kompetens.

För varje extra kompetens ser man också att användningen blir högre vilket är logiskt då fler samtal blir betjänade men inga fler agenter finns och den genomsnittliga betjäningstiden är densamma måste fler agenter arbeta mer för att ta hand om de samtal som inte blir blockerade.



Figur 1: Andel blockerade samtal, ASA, SL och användning för 3 olika tryck och 1-6 kompetenser där genomsnittlig betjäningstid är 10 minuter för alla kompetensnivåer.

5.1.1 Matematisk beräkning av ett callcenter

Med hjälp av Erlang-X-formeln är det möjligt att räkna ut SL, ASA och sannolikhet att ett samtal blockeras för ett callcenter med bara en kompetens. Callcentret med sex kompetenser som simulerades i sektion 5.1 agerar på samma sätt som ett callcenter med bara en kompetens och bör få liknande värden som beräkningen. Indata i Erlang-X-kalkylatorn Koole är 10 minuter för betjäningstid, 90 agenter, 120 telefonlinjer och 7.74, 8.4 respektive 9

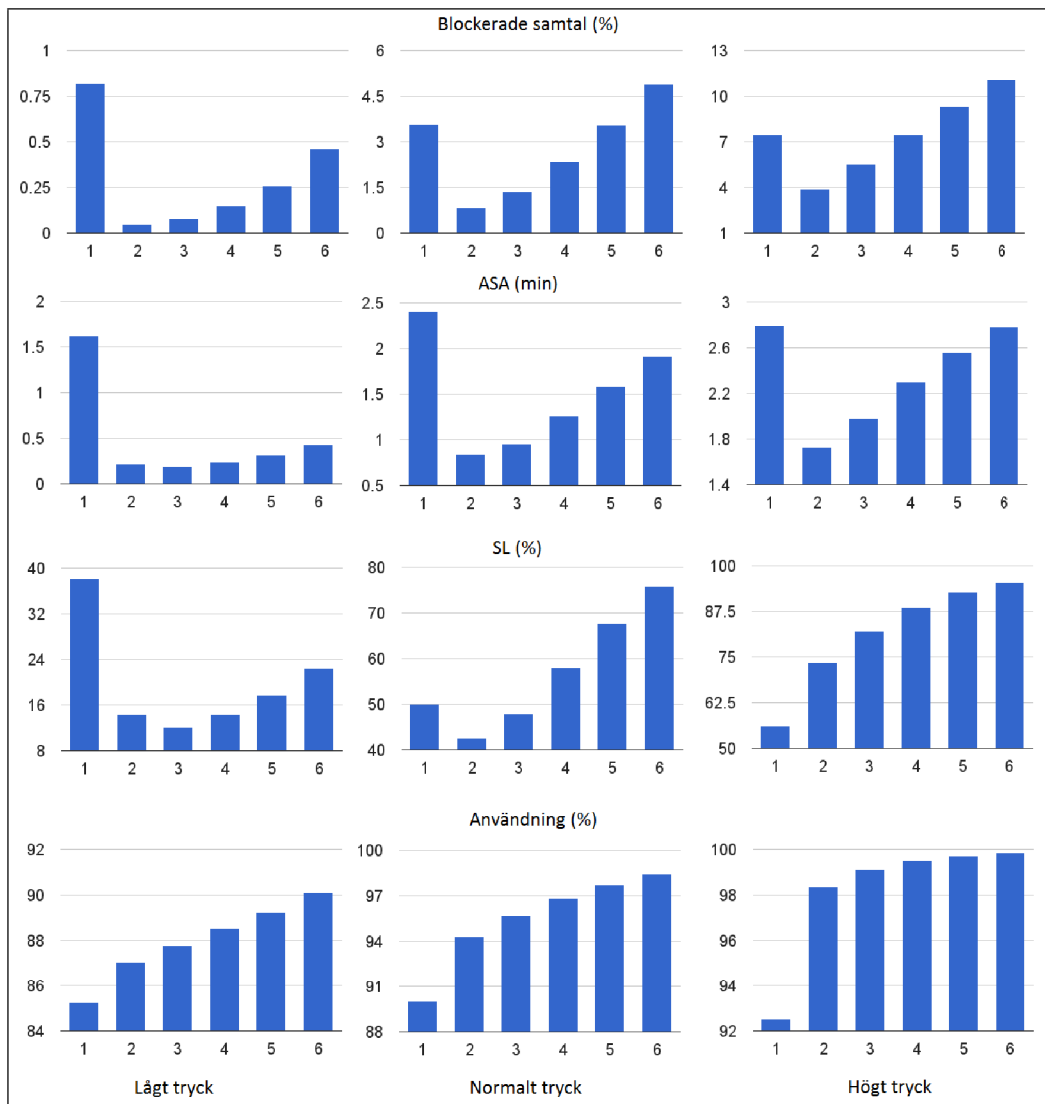
ankomster per minut för lågt, normalt och högt tryck. Resultatet av beräkningen syns i tabell 1 tillsammans med resultaten från simuleringen med sex kompetenser per agent som visas i figur 1. Som synes följer resultatet från simuleringen beräkningens resultat mycket väl.

Tryck	Typ	Blockerad (%)	ASA (min)	SL (%)
Lågt	Simulering	0.017	0.082	5.77
	Beräkning	0.02	0.083	5.77
Normalt	Simulering	0.35	0.45	26.56
	Beräkning	0.36	0.45	26.71
Högt	Simulering	2.34	1.24	61.39
	Beräkning	2.35	1.24	61.35

Tabell 1: Jämförelse mellan beräknade värden med hjälp av Erlang-X-kalkylatorn [Koole](#) för ett callcenter med 90 agenter, 120 telefonlinjer, betjäningstid 10 minuter och 7.74, 8.4 respektive 9 ankomster per minut med simulerade värden för ett callcenter där alla agenter har alla kompetenser och den genomsnittliga betjäningstiden är 10 minuter för alla kompetensnivåer.

5.2 Simulering med liten skillnad av genomsnittlig betjäningstid

Här simuleras ett callcenter med 5 % skillnad i genomsnittlig betjäningstid mellan varje kompetensnivå. Trots att det tar längre tid för en agent att ta hand om ett samtal med andra kompetenser än dennes primära kompetens tjänar man på att inför flera kompetenser för alla typer av callcenter. För fler kompetenser än två är den generella trenden att effektiviteten försämras för varje kompetens. Högt tryck ger bäst resultat för två kompetenser om man bortser från SL där möjligheten för en agent att svara på fler samtalstyper gör att färre samtal blockeras och att kön liknar mer en FIFO-kö. Alltså fås bäst resultat med två kompetenser. Se figur 2.

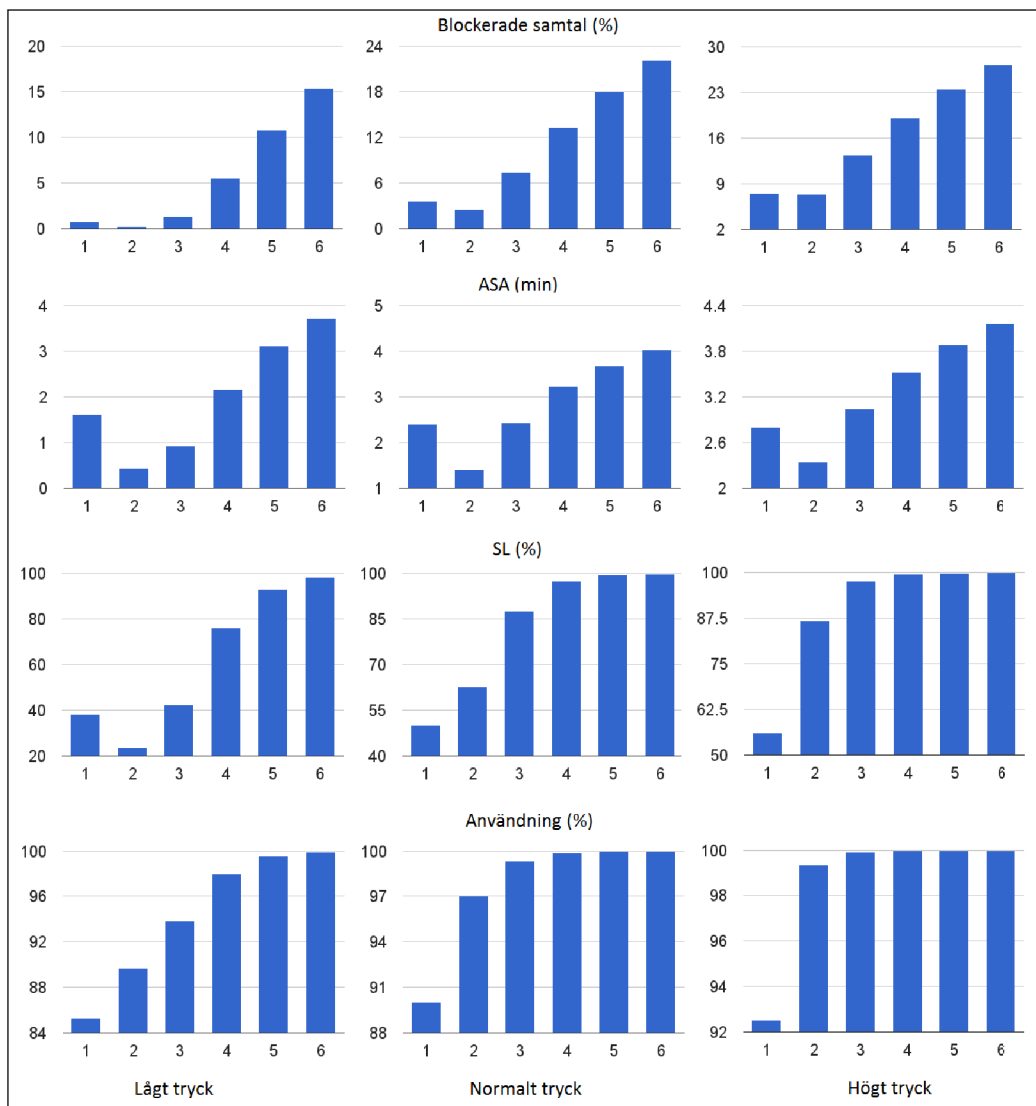


Figur 2: Andel blockerade samtal, ASA, SL och användning för tre olika tryck och 1-6 kompetenser där genomsnittlig betjäningstid beräknas enligt formel (2) för $n = 0.05$ för alla kompetensnivåer X .

5.3 Simulering med stor skillnad av genomsnittlig betjäningstid

Även när skillnaden i genomsnittlig betjäningstid är 15 % mellan kunskapsnivåerna fås ett resultat som liknar det med 5 % skillnad. Resultatet visas i figur 3. Det som skiljer sig är att förbättringen mellan en kompetens och två kompetenser är mycket mindre. Dessutom är det i de flesta fallen det callcentret med två kompetenser som är det ända som är bättre än call-

centret med endast en kompetens. SL försämras också för högt tryck som simuleringen i figur 2 men här är den även sämre för normalt tryck.



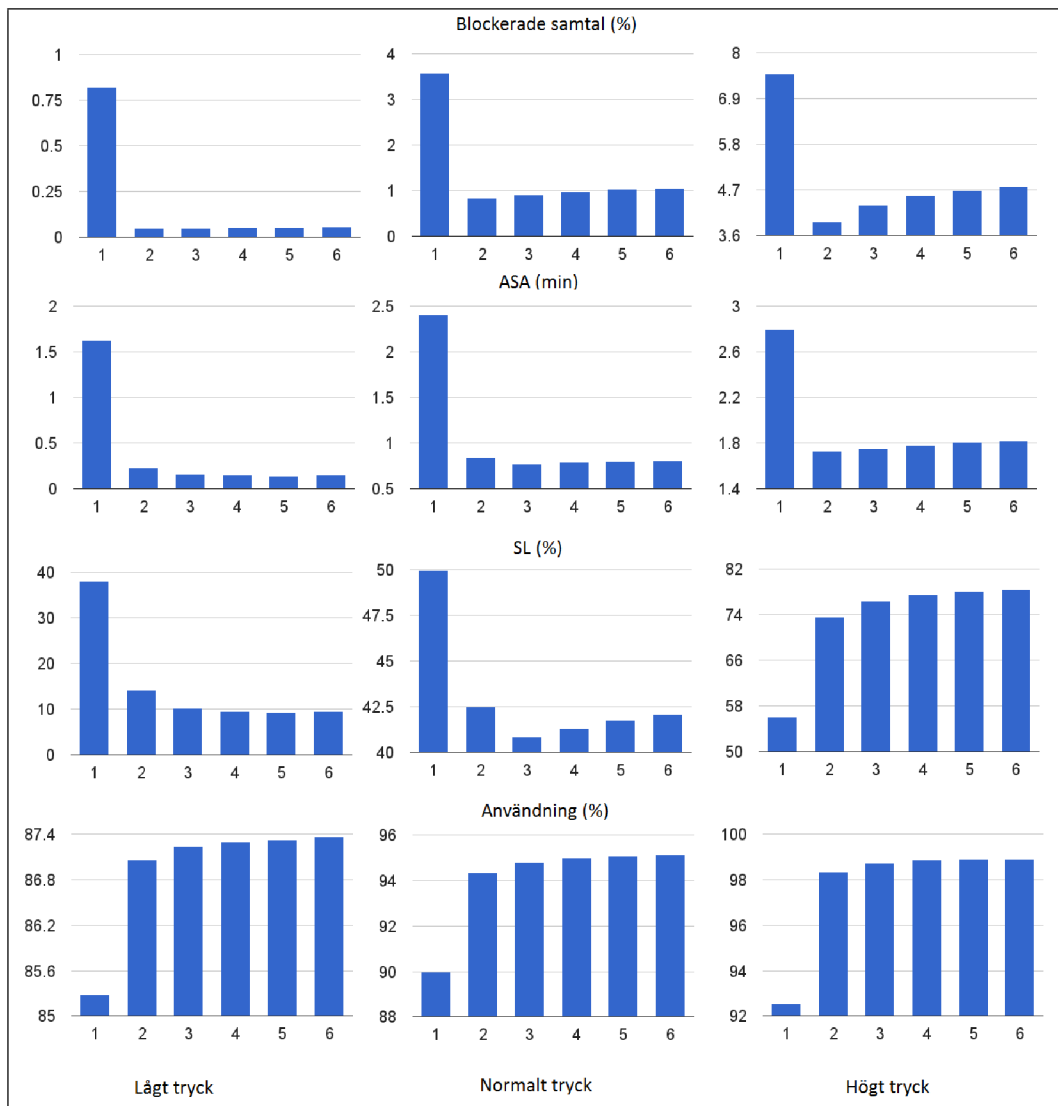
Figur 3: Andel blockerade samtal, ASA, SL och användning för tre olika tryck och 1-6 kompetenser där genomsnittlig betjäningstid beräknas enligt formel (2) för $n = 0.15$ för alla kompetensnivåer X .

5.4 Simulering med en primär kompetens och övriga sekundära

I simuleringarna, vars resultat visas i figur 2 och 3, fås det bästa resultatet när varje agent har två kompetenser. Det skulle kunna bero på att två kompetenser ger de styrkor som ett multi-skill callcenter har då fler agenter kan ta hand om flera typer av samtal medan att använda tre kompetenser endast tillför längre betjäningstid i dessa simuleringar. Figur 1 som simulerade ett callcenter med fixerad genomsnittlig betjäningstid på 10 minuter talar också för det då den simuleringen visar att den största effektivitets-skillnaden ligger i att använda två kompetenser istället för en, medan tre kompetenser istället för två ger marginella skillnader. För att kontrollera detta simuleras ett callcenter där den primära kompetensen har genomsnittlig betjäningstid 10 minuter och alla övriga sekundära kompetenser har genomsnittlig betjäningstid 10,5 minuter precis som den andra kompetensen i figur 2. Resultatet visas i figur 4.

När det är lågt tryck stämmer hypotesen eftersom blockerade samtal, ASA och SL blir bättre för varje kompetens som införs, även om förbättringarna är små. För normalt tryck gäller att tre kompetenser ger bäst resultat för ASA och SL medan andel blockerade samtal blir som lägst för två kompetenser.

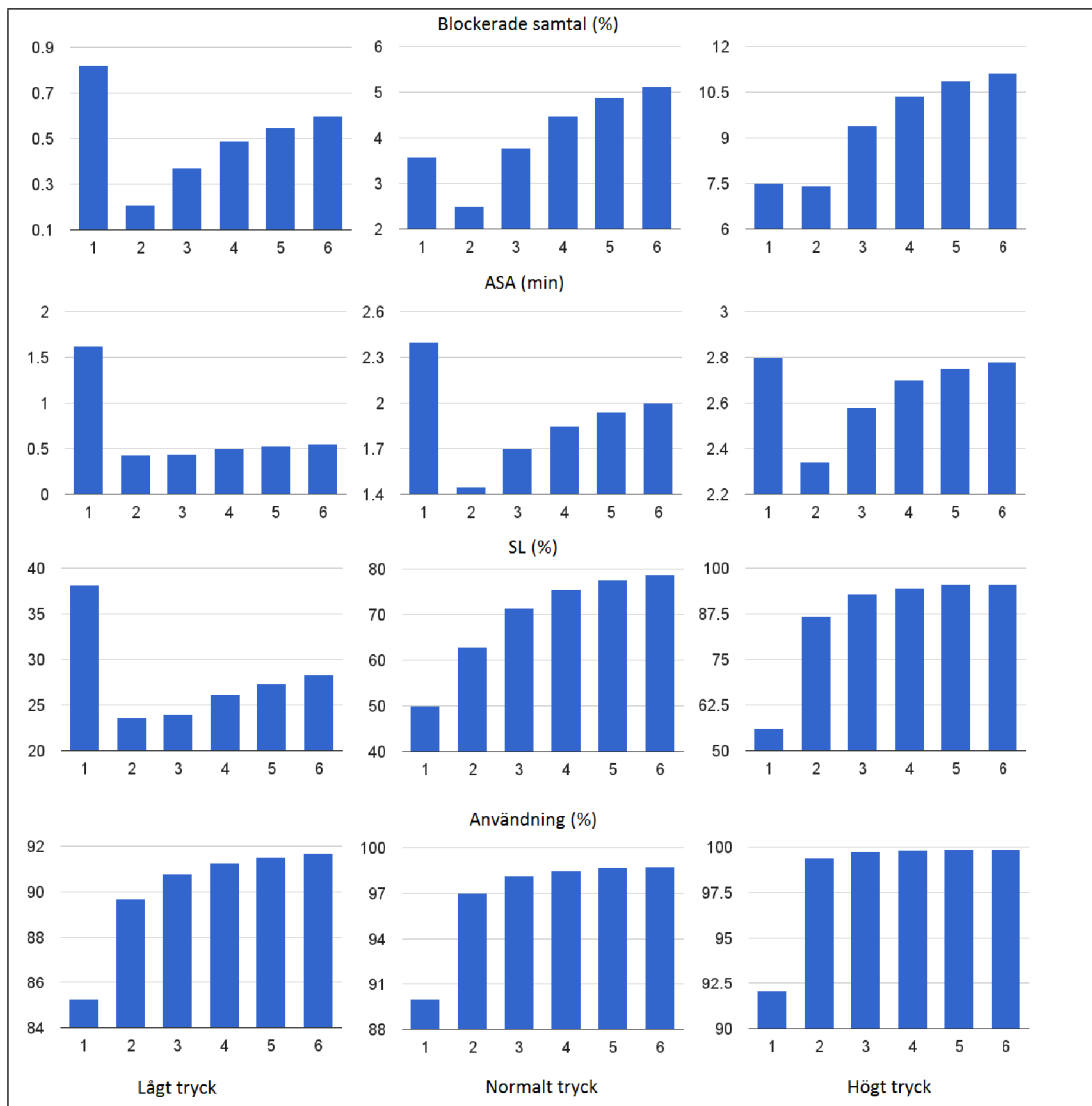
När de sekundära kompetenserna har genomsnittlig betjäningstid 11,5 minuter som den andra kompetensen i figur 3 blir resultatet tydligare. Där visar sig två kompetenser fungera betydligt bättre än tre kompetenser för normalt och högt tryck. Även en lite skillnad finns för lågt tryck till fördel för två kompetenser. se figur 5



Figur 4: Andel blockerade samtal, ASA, SL och användning för tre olika tryck och 1-6 kompetenser där genomsnittlig betjäningstid är 10 minuter för den primära kompetensen och 10,5 för övriga kompetenser.

6 Slutsats

För ett multi-skill callcenter utan varierande genomsnittligt betjäningstid har det visats att callcentret blir effektivare för varje extra kompetens agenterna får. Med varierande genomsnittlig betjäningstid har det visat sig att det inte stämmer. För tidsskillnader på 5 % och 15 % i genomsnittlig betjäningstid finns en tydlig trend som visar att effektiviteten först blir bättre men sen blir sämre i takt med att antal kompetenser som används ökas. Till slut blir



Figur 5: Simulerade värden för tre olika tryck och 10 minuter genomsnittlig betjäningstid för agenters primära kompetens, 11.5 minuter för övriga.

effektiviteten lika dålig eller sämre än då callcentret bara har en kompetens per agent. För skillnaden 5 % går det långsamt där den första parametern som får samma värde eller högre än för 1 kompetens är andelen blockerade samtal när agenterna har fyra kompetenser var. För skillnaden 15 % går det betydligt snabbare där användningen av tre kompetenser per agent ger en effektivitet som genomgående är sämre än då en kompetens per agent används. Även om skillnaderna 5 % och 15 % i genomsnittlig betjäningstid används så att den primära kompetensen har genomsnittlig betjäningstid 100 % och resterande 105 % respektive 115 % fås ett liknande resultat. Det

leder till antagandet att givet att man inte kan garantera att alla agenter blir lika kompetenta på alla kompetenser de lär sig, bör man inte försöka träna agenter till att kunna ta hand om alla samtal.

Alla simuleringar med någon form av varierande genomsnittlig betjäningstid visade på ett specifikt antal kompetenser per agent som gav bäst resultat. I de flesta fallen var det när två kompetenser per agent användes. I de få fall där det inte gällde var det användningen av tre kompetenser per agent som var bäst där skillnaden med användningen av två kompetenser var liten eller försumbar. Alltså är användningen av två kompetenser per agent det effektivaste antalet kompetenser eller så är det nära det effektivaste resultatet för callcentret som simulerades.

SL skilde sig något från övriga faktorer i den mening att när andel blockerade samtal och ASA blev mindre ökade SL för stor skillnad i genomsnittlig betjäningstid och/eller högt tryck. Det blir ett tydligt byte mellan fler samtal som blir betjänade och samtal som får vänta längre än gränsen för SL. För ett callcenter som prioriterar SL över andel samtal som blir betjänade bör endast användas en kompetens.

Slutligen kunde man också se att med fler kompetenser blev användningen högre. Stor skillnad i genomsnittlig betjäningstid gjorde att den blev högre än med låg skillnad i genomsnittlig betjäningstid. I vissa extrema fall närmade sig användningen 100 % vilket är orealistiskt om man endast räknar in samtalstiden i betjäningstiden. Dessutom gäller det endast för fler än två kompetenser per agent vilket inte är effektivt att använda som redan diskuterats.

Om man inte kan garantera likvärdiga prestationer tidsmässigt per samtalstyp givet en agents kompetenser bör två kompetenser per agent användas. Även om man anser att agenterna i ett callcenter har samma genomsnittliga betjäningstid för olika typer av samtal tjänar man inte så mycket på att använda mer än två kompetenser men faran att få sämre effektivitet finns eftersom endast små skillnader i genomsnittlig betjäningstid ger negativa resultat för fler än två kompetenser.

Referenser

- [Anton, Babat och Paul 1999] Anton, J., V. Bapat, Hall, B. (1999) *Call Center Performance Enhancement Using Simulation and Modelling*. Purdue University Pres. Kapitel 2. Tillgänglig på http://books.google.se/books?id=EKcyyIBiCkC&printsec=frontcover&hl=sv&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- [Engel och Grandell, 2006] Enger, J. Grandell, J (2006) *Markovprocesser och köteori*. Tillgänglig på <http://www.math.kth.se/matstat/gru/markovkomp.pdf>
- [Koole] Koole, G. *Erlang X calculator*. Tillgänglig på <http://www.gerkoole.com/CC0/ErlangX.php>
- [Koole, 2007] Koole, G. (2007) *Call center mathematics*. Tillgänglig på <http://www.cs.vu.nl/~koole/ccmath/book.pdf>
- [Wallace och Whitt, 2005] Wallace, R. Whitt, W. (2005) *A staffing algorithm for call centers with skill-based routing*. *Manufacturing and service operations management*, Vol 7, issue 4, s 276-294. Tillgänglig på <http://pages.stern.nyu.edu/~gjanakir/WhittPaper.pdf>
- [Whitt, 2005] Whitt, W. (2005) *Engineering solution of a basic call-center modell*. Tillgänglig på <http://www.columbia.edu/~ww2040/EngineeringPub.pdf>
- [Winter, 2004] Winter. (2004) *Discrete Event Simulation*. Tillgänglig på http://engineering.dartmouth.edu/~Eric_W_Hansen/engs027/outlines/19DiscrEvSim.pdf

A Invertering av exponentialfunktionen

Exponentialfunktionen används för att räkna ut sannolikheten U för att det sker en händelse under en tidsperiod X för en given genomsnittligt tidsperiod λ . Inversen räknar ut en tidsperiod givet en sannolikhet där alla sannolikheter $0 - 1$ är lika sannolika vilket gör att det går att använda en likafördelad slumpgenerator för att generera U . Notera att när U antar slumpmässiga likafördelade värden mellan 0 och 1 är $1 - U$ och U ekvivalenta.

$$\begin{aligned}U &= 1 - e^{-X/\lambda} \\ \Rightarrow \\ X &= -\lambda * \ln(1 - U) \\ \Rightarrow \\ X &= -\lambda * \ln(U) \end{aligned} \tag{1}$$

