

Beroendeanalys ur ett flödesperspektiv

– Utveckling av analysverktyg

Einar Améen

Axel Andersson

**Department of Fire Safety Engineering and Systems Safety
Lund University, Sweden**

**Brandteknik och Riskhantering
Lunds tekniska högskola
Lunds universitet**

Report 5435, Lund 2013

Beroendeanalys ur flödesperspektiv

– utveckling av analysverktyg

**Einar Améen
Axel Andersson**

Lund 2013

Beroendeanalys ur flödesperspektiv – utveckling av analysverktyg
Dependency analysis from a flow perspective – the development of an analysis tool

Einar Améen & Axel Andersson

Report 5435

ISSN: 1402-3504

ISRN: LUTVDG/TVBB--31XX--SE

Number of pages: 137

Illustrations: Einar Améen and Axel Andersson

Key words

Urban Flows, Aggregating dependency, Computer tool, Dependency analysis, Risk and vulnerability analysis.

Nyckelord

Flöde, Sammanslagning av beroenden, Datorverktyg, Beroendeanalys, Risk- och sårbarhetsanalys.

Abstract

In today's modern society, there is a trend that different societal functions are more interconnected, both by the use of technology and by shared service providers. The amount of connections in the societal system makes it impossible, without a tool, to take all consequential effects of a perturbation in consideration. The concept of urban flows is previously developed to help resolve this issue and it is further developed in this thesis. The goal of the thesis was to create a calculation methodology that aggregate dependencies and influence from actors to urban flows on a municipal level. The developed calculation methodology is based on assumptions and the accuracy of the results is limited by these assumptions. A MATLAB based computer tool was developed to aggregate dependencies and influence according to the calculation methodology and visualise the results. A small case study that proves the value of the computer tool in a risk- and vulnerability analysis is also presented.

© Copyright: Brandteknik och Riskhantering, Lunds tekniska högskola, Lunds universitet, Lund 2013.

Brandteknik och Riskhantering

Lunds tekniska högskola

Lunds universitet

Box 118

221 00 Lund

Department of Fire Safety Engineering

and Systems Safety

Lund University

P.O. Box 118

SE-221 00 Lund

Sweden

Sammanfattning

I dagens moderna samhälle har utvecklingen gått mot att olika samhällsfunktioner blivit mer sammankopplade, bland annat genom ökad användning av teknik och gemensamma tjänsteleverantörer. Det bildas komplexa kopplingar som gör att en viktig samhällsfunktion kan påverkas starkt av att andra aktörer störs, även om direkta kopplingar mellan aktörerna saknas. Svåridentifierade kopplingar som dessa gör att det krävs väl utvecklade arbetsmetoder för att öka robustheten i samhället.

Genom kartläggning av beroenden kan olika aktörer i samhället på ett systematiskt sätt öka sin robusthet. Detta eftersom en kartläggning kan användas för att förutsäga och därefter minska effekterna av en störning. Att identifiera de mest kritiska beroendena kan vara svårt eftersom det ofta finns ett stort antal beroenden samt att dessa kan vara indirekt kopplade till varandra. En sätt att underlätta arbetet kring beroendeanalys är begreppet flöden. Begreppet har tagits fram av Resilient Regions Association och vidareutvecklats på LTH av J. Johansson och Hassel samt genom ett examensarbete av H. Johansson och Åhsberger (2012). Flöden beskrivs som förflyttningen av varor, tjänster, människor, energi, kapital eller information. Genom att använda flödesbegreppet går det att förmedla ett beroende mellan olika aktörer. Ett exempel är flödet dricksvatten som utgår från vattenverket och går till användarna samtidigt som det påverkas av tillsyn från miljöförvaltningen.

Den stora fördelen med flödesbegreppet, när väl innebörden av ett flöde är klargjord, är att det på ett förhållandevis enkelt sätt går att identifiera kopplingar mellan olika aktörer. Att klargöra vad ett flöde är och skapandet av en samstämmig bild av ett specifikt flöde är däremot något som bör förbättras ytterligare. Enligt den tidigare utvecklade metodiken sker identifiering av kopplingar mellan aktörer och flöden genom att aktörerna får ange styrka på både beroende och påverkan för varje flöde. Datainsamling utförs i form av workshops med expertbedömningar, vilket gör att det även går att identifiera kopplingar till flöden som saknar fysisk infrastruktur eller sådana som inte är lika tydliga i till exempel ekonomisk data. Ytterligare en fördel med flödesbegreppet är att det möjliggör analys av beroende och påverkan som förändras över tiden. Detta är möjligt eftersom aktörerna anger sina flödeskopplingar i tidsperioder från 0-2 timmar upp till år.

Manuellt går det inte att aggregera och analysera insamlad data eftersom de undersökta systemen snabbt blir för stora. Examensarbetet har därför syftat till att utveckla en beräkningsmetod och ett datorverktyg som möjliggör analys av beroenden mellan aktörer genom flöden. Ett första steg har varit en litteraturstudie där befintliga modeller och metoder beskrivits. Parallellt har det skapats en kravspecifikation för datorverktyget tillsammans med potentiella användare från Lunds kommun. Litteraturstudien och kravspecifikationen har lett fram till slutsatsen att befintliga modeller och metoder inte går att tillämpa rakt av och att en egen beräkningsmetod ska utvecklas. Beräkningsmetoden har utvecklats och implementerats i ett datorverktyg skapat i MATLAB. Stort fokus har legat på att jämföra och motivera de antaganden som legat till grund för beräkningsmetoden.

En begränsad fallstudie har genomförts på Lunds kommun, både för att undersöka datorverktygets möjligheter som hjälpmedel för beroendeanalys och för att identifiera specifika för- och nackdelar med datorverktyget och flödesbegreppet. Fallstudien visade att det på ett användbart sätt gick att kvantifiera beroenden och även att identifiera indirekta beroenden. Baserat på fallstudien och övriga delar av examensarbetet har det tagits fram ett antal förslag på fortsatt arbete. De fyra viktigaste områdena är utökad datainsamling, utvärdering av antagandet att påverkan sker linjärt, införandet av en ny aktör kallad privatpersoner och att validera beräkningsmetoden mot andra modeller eller mot verkliga händelser.

Summary

In today's modern society, there is a trend that different social functions are more interconnected, both by the use of technology and by shared service providers. Complex networks are created within the societal system and these networks could make an important social function vulnerable to a large number of actors. The vital social function could be vulnerable to actors even without direct connections between them. Connections like these are difficult to identify and they require systematic methods to improving the overall societal resilience.

By mapping dependencies in a systematic way it is possible to increase the resilience of many actors important to the society. This is because the mapping can be used to predict and reduce the effects of a perturbation. Identifying the most critical dependencies can be difficult, mainly because there are often a large number of dependencies and these may be indirectly connected. One way to ease the work with dependency analysis is the concept of urban flows developed by Resilient Regions Association, further developed at LTH by J. Johansson and Hassel and as a master thesis by H. Johansson and Åhsberger (2012). Urban flow is described as the movement of goods, services, people, energy, capital or information. By using the concept of urban flows it is possible to link a dependency between different actors. One example of an urban flow is drinking water, which starts at the water plant and goes to the users and at the same time being affected by supervision from the environmental administration.

The major advantage of the urban flow concept is that, once the meaning of urban flows is clarified, it is relatively simple to identify connections between different actors. However, further work is needed to clarify the urban flow concept and create a consistent definition of specific urban flows. According to the previously developed methodology, the connections between actors and urban flows are defined by the actors during workshops. The actors indicated the strength of their dependency and influence on the different urban flows. The data collection is performed in terms of expert estimates, which means that it is also possible to identify links to urban flows that lack physical infrastructure, or those that are not obvious in for instance economic data. Another advantage of the urban flow concept is that it allows analysis of the dependency and influence that change over time. This is possible because the actors indicate their urban flow connections in time periods from 0-2 hours up to years.

It is not possible aggregate and analyse the collected data manually since the systems studied quickly becomes too large to handle. The goal of this thesis has therefore been to develop a calculation methodology and computer tool that enables analysis of dependencies between actors through urban flows. The first step has been a literature review where existing models and methods have been described. Simultaneously and together with potential users from municipality of Lund, a requirement specification has been created for the computer tool. The literature review and the requirement specification led to the conclusion that the existing models and methods were not suited for the work with urban flows and that a new calculation methodology should be developed. The calculation methodology was developed and thereafter implemented in a computer tool created in MATLAB. A major effort has been to compare and justify the assumptions underlying the calculation methodology.

A limited case study has been conducted on the municipality of Lund, both to investigate the potential of the computer tool as an instrument for dependency analysis and to identify specific pros and cons of the computer tool and urban flow concept. The case study showed that the tool was useful both to quantify dependencies and to identify indirect dependencies. Based on the case study and other parts of the thesis there is a number of suggestions for further work. The four main areas are extended the data collection, evaluation of the assumptions that influence occurs linearly, the introduction of new actor called individuals and to validate the calculation methodology with other models or with actual events.

Förord

Med detta examensarbete avslutar vi vår utbildning vid Lunds Tekniska Högskola. Efter många års studier är vi äntligen redo för att möta en spännande arbetsmarknad.

Nedanstående personer har varit av stor betydelse för att vi skulle kunna genomföra detta arbete och vi vill därför rikta ett stort tack till er.

Först och främst vill vi tacka våra fantastiska handledare **Jonas Johansson** (forskarassistent vid Avdelningen för Industriell Elektroteknik och Automation, LTH) och **Linn Svegrup** (doktorand vid Avdelningen för Brandteknik och Riskhantering, LTH) för värdefulla handledarmöten och stort engagemang som bland annat visat sig i de 1281 skriftliga kommentarerna som vi har fått. Utan ert stöd och engagemang hade aldrig detta examensarbete lett fram till denna rapport och den beräkningsmodell som vi nu är stolta över att få lämna ifrån oss.

Vi vill även rikta ett stort tack till **Henrik Hassel** (universitetslektor vid Avdelningen för Brandteknik och Riskhantering, LTH) för stödet i uppstarten av examensarbetet.

Vidare vill vi tacka **Eva Greko** och **Lennart Larsson** (säkerhetssamordnare, Lunds kommun) för hjälpen med att skapa den kravspecifikation som vi har arbetat mot.

Vi vill även rikta ett tack till Resilient Regions Association och **Olle Uhlman** samt iFACTS och **Stefan Schreiter** för deras stöd.

Slutligen vill vi även tacka **våra underbara vänner i Ateljé Q** (vårt exjobbsrum) för stöd och glada kommentarer under arbetets gång men framförallt alla trevliga luncher och goda hembakade fredagsfikan.

Lund 2013-11-28



Einar Améen



Axel Andersson

Innehållsförteckning

1 Inledning.....	1
1.1 Bakgrund.....	1
1.2 Syfte	2
1.3 Mål	2
1.4 Målgrupp.....	3
1.5 Avgränsningar	3
1.6 Plan för genomförande.....	3
1.6.1 Etapp 1 – Bakgrundskunskap.....	4
1.6.2 Etapp 2 – Specificera syfte.....	4
1.6.3 Etapp 3 – Designkriterier; identifiera indata och efterfrågade resultat.....	4
1.6.4 Etapp 4 – Ta fram lämplig beräkningsmetod	4
1.6.5 Etapp 5 – Tillämpa beräkningsmetoden och skapa datorverktyg.....	5
1.6.6 Etapp 6 – Analys av metod och verktyg.....	5
1.6.7 Etapp 7 – Lärdomar från analys och användande.....	5
2 Teoretisk bakgrund.....	7
2.1 Riskhantering	7
2.2 Risk- och sårbarhetsanalys	8
2.3 Flöden	9
2.4 Beroende.....	9
2.4.1 Indirekt beroende	10
2.5 Stötdämpare	10
2.6 Modeller och verktyg för beroendeanalys	11
2.6.1 Nätverksmodeller.....	11
2.6.2 Input-output-modeller.....	12
2.6.3 System dynamics.....	13
2.6.4 Agentbaserade modeller	13
2.6.5 Manuella undersökningsmetoder och kvalitativa rapporter.....	14
2.6.6 Supply Chain Risk Management.....	14
2.6.7 Summering av tillgängliga modelleringsalternativ	15
3 Metod.....	17
3.1 Metod för användarintervju.....	17
3.2 Metod för utveckling av datorverktyg.....	17
3.3 Validering och verifiering.....	17
4 Empirisk data.....	21

5 Utveckling av beräkningsmetod och datorverktyg	23
5.1 Antagande.....	23
5.1.1 Geografiska antaganden	23
5.1.2 Analytiska antaganden.....	23
5.1.3 Antaganden kopplade till tillgänglig indata.....	24
5.2 Metoder för att beräkna beroende och påverkan.....	26
5.2.1 Beräkningsmetod 1 - Additionsmetoden	27
5.2.2 Beräkningsmetod 2 - Multiplikationsmetoden.....	28
5.2.3 Beräkningsexempel.....	29
5.2.4 Summering och val av beräkningsmetod.....	32
6 Visualisering av resultatet.....	35
6.1 Visualiseringar av indata	35
6.1.1 Direkta kopplingar mellan aktörer via flöde.....	35
6.1.2 Direkta kopplingar mellan flöden	36
6.1.3 Direkt flödespåverkan och flödesberoende för en aktör	36
6.1.4 Samtliga aktörers beroenden och påverkan under en tidsperiod	37
6.1.5 Spridning- och beroendekedjor för en aktör.....	38
6.2 Visualisering av analysresultat	39
6.2.1 Aktörers fullständiga beroende av flöden.....	39
6.2.2 Matris över aktörers beroende av flöden	40
6.2.3 Matris över flödens beroende av övriga flöden	40
6.2.4 Matris över aktörernas beroende av övriga aktörer	41
6.2.5 De viktigaste aktörerna och flödena för valfri aktör och kommun.....	42
6.2.6 Spridningskedjor från en aktör till alla påverkade aktörer	43
6.2.7 Responsgraf	44
6.3 Verifiering.....	45
6.3.1 Verifiering av hur aktörer beror av andra aktörer	46
6.3.2 Verifiering av hur aktörer beror av flöden.....	47
6.3.3 Verifiering av påverkan från aktörer med nedsatt kapacitet	48
7 Fallstudie – Lunds kommun.....	51
7.1 Antagande.....	52
7.2 Resultat och inledande analys	52
7.3 Figurer som skapats men inte använts i fallstudien	62
7.4 Diskussion - Fallstudie	63
7.4.1 Diskussion kring antaganden.....	64
7.4.2 Övrig diskussion av fallstudien.....	64

8	Diskussion	65
8.1	Generella lärdomar från projektet	65
8.2	Diskussion kring validering och verifiering.....	65
8.3	Diskussion kring indata	66
8.3.1	Brister i befintlig indata.....	66
8.3.2	Flöden och geografisk information	67
8.4	Utvärdering av beräkningsmetod.....	68
8.5	Utvärdering av datorverktyg.....	70
8.6	Generell utvärdering av flödesbegreppet.....	70
9	Slutsats	73
9.1	Förslag på fortsatt arbete	73
9.1.1	Fortsatt utveckling av datorverktyget.....	74
9.1.2	Förslag till fortsatt arbete kring beroendeanalys genom flöden.....	74
10	Litteraturförteckning	75
Appendix A.	Kravspecifikation.....	A-1
Appendix B.	Funktionskarta för MATLAB-koden.....	B-1
Appendix C.	Exempel på indata	C-1
Appendix D.	MATLAB-kod för datorverktyget.....	D-1

1 Inledning

Examensarbetet är den avslutande delen på utbildningen till Civilingenjör i Riskhantering och Brandingenjör vid Lunds tekniska högskola (LTH). Detta examensarbete skrevs hösten 2013.

1.1 Bakgrund

Den 8 januari 2005 drog stormen Gudrun in över Halland med full styrka, stormen kom att skada upp till 30 000 km elledningar i södra Sverige (KBM, 2005). Vagnätets funktion var kraftigt nedsatt, vilket var en bidragande orsak till att det tog upp till 40 dygn innan alla kunder åter hade elektricitet. Även telenätet blev hårt drabbat vilket gjorde samordning mellan olika aktörer svårt. Viktiga samhällsfunktioner som hemtjänst blev därmed kraftigt påverkat, både av den begränsade framkomligheten och av det skadade telenätet. Enskilda aktörer och verksamheter var ofta beroende av flera andra aktörer, vilket i vissa fall innebar cirkelberoenden.

Stormen Gudrun visade hur störningar i ett system fick både stora väntade och oväntade följd effekter i andra system (KBM, 2005). Genom att kartlägga beroenden går det att undvika att vissa följd effekter blir oväntade (Rinaldi, Peerenboom, & Kelly, 2001). Kommuner är enligt MSBFS 2010:6 ålagda att identifiera dessa kritiska beroenden som en del av deras risk- och sårbarhetsanalyser. Att kommuner ska utföra risk- och sårbarhetsanalyser är lagstadgat i SFS (2006:544), *om kommuners och landstings åtgärder inför och vid extraordinära händelser i fredstid och höjd beredskap*. Vad en risk- och sårbarhetsanalys innebär, och därmed beroendeanalys, beskrivs i MSBFS 2010:6. Analyser av beroenden är därmed en förhållandevis ny del av det svenska risk- och sårbarhetsarbetet och det finns ett stort behov av metodutveckling inom området (J. Johansson, personlig kommunikation, 16 september, 2013).

För att identifiera sårbarheter i samhället kan ett tillvägagångssätt vara att analysera beroende i form av flöden. Flöden är ett nytt begrepp inom den svenska forskningen kring kritisk infrastruktur och samhällsviktig verksamhet. H. Johansson och Åhsberger (2012) beskriver flöden som förflyttningen av varor, tjänster, människor, energi, kapital eller information. Flöden förmedlar ett beroende mellan olika funktioner, ett exempel är flödet vatten som utgår från vattenverket och går till användarna samtidigt som det påverkas av tillsyn av miljöförvaltningen. Det är i första hand intressant att studera flöden kopplat till de funktioner som är så kallade samhällsviktiga. MSB (2011a) definierar samhällsviktig verksamhet som:

”En samhällsfunktion av sådan betydelse att ett bortfall av eller en svår störning i funktionen skulle innebära stor risk eller fara för befolkningens liv och hälsa, samhällets funktionalitet eller samhällets grundläggande värden.” (MSB, 2011a, s.10)

I dagsläget diskuteras och utvecklas analyser av ömsesidigt beroende infrastruktur, som bland annat skulle kunna användas för sårbarhetsanalys (se till exempel Rinaldi, Peerenboom, & Kelly, 2001). Ett alternativ till att analysera infrastruktur kan vara att analysera flöden. Skillnaden mellan infrastruktur och flöden är att infrastruktur är fysiska strukturer i samhället, som elnätet och vattennätet, som transporterar flödena elektricitet och vatten. Anledningen till att det är användbart att studera flöden är för att flöden enklare kopplas till samhällsviktig verksamhet och för att flöden kan hantera fler aspekter än enbart fysiska kopplingar (H. Johansson & Åhsberger, 2012).

Arbetet kring kritiska flödesberoenden kommer från ett samarbete mellan LTH och Resilient Regions Association (tidigare Training Regions) och examensarbetet tar sin utgångspunkt i detta samarbete. En del av detta samarbete är metodutveckling för att kartlägga och analysera flöden, vilket resulterade i ramarna för en metod (J. Johansson & Hassel, personlig kommunikation, 16 september, 2013). Under hösten 2012 skrevs ett examensarbete av H. Johansson och Åhsberger (2012) som ingående beskriver teorin kring

flöden samt beroenden och olika metoder för datainsamling. H. Johansson och Åhsbergers rapport används som utgångspunkt för detta examensarbete och data insamlad av H. Johansson och Åhsbergers ligger till grund för beräkningsmetoden och datorverktyget som utvecklats inom ramen för detta examensarbete.

Den data som H. Johansson och Åhsberger (2012) samlade in består bland annat av flödesberoende och flödespåverkan baserat på de aktiviteter som de olika aktörerna bedriver. Resultatet av datainsamlingen bildar ett komplext nätverk med flertalet beroenden mellan aktörer. Ordet nätverk syftar här till en struktur med noder som är sammankopplade med länkar, vilket kan beskriva både människors interaktioner och infrastruktur (Johansson, Jönsson, & Johansson, 2007). Med hjälp av nätverksteori kan påfrestningar i dessa komplexa system analyseras och sårbarheter utredas och utvärderas.

En metod som från början utvecklats för att förutse den nationalekonomiska framtiden, men som har modifierats och numera även används för att beroendeanalys, är den så kallade Leontief input-output-modellen (Haines & Jiang, 2001). Modellen beskriver hur olika sektorer i en ekonomi interagerar med hjälp av vilka monetära belopp som förflyttas mellan de olika sektorerna, men den kan även användas för att uppskatta beroenden och effekten av dessa beroenden. Haines och Jiang (2001) har anpassat Leontiefs input-output-modell till riskhantering av sammanlänkade sektorer i form av Inoperability Input-output Model. Input-output-modeller har använts för flertalet olika former av beroendeanalys (Ginstrup & Rubil, 2013; Haines & Jiang, 2001; H. Johansson, Svegrup, & Hassel, 2013; Setola & De Porcellinis, 2008) och i flera av dessa analyser finns svar på liknande frågeställningar och problem som finns vid hantering av flödesberoenden.

1.2 Syfte

Examensarbetet syftar till att på sikt hjälpa kommuner och län i arbetet att bli mer robust mot störningar i samhällsviktiga verksamheter. För att underlätta detta arbete är det specifika syftet här att utveckla en beräkningsmetod och ett datorverktyg för att analysera beroenden mellan aktörer genom flöden. Förhoppningen är att kommuner och län ska få bättre möjligheter att göra väl grundade bedömningar av beroenden baserat på flöden och att därefter kunna använda bedömningarna som en delmängd av sitt risk- och sårbarhetsarbete.

1.3 Mål

Huvudmålet är att skapa en beräkningsmetod för att aggregera och analysera flödesdata. För att utföra analyser i stor skala är målet att en prototyp av ett datorverktyg ska utvecklas som kan analysera flödesberoenden på en aggregerad nivå. Med en aggregerad nivå avses en sammanställning av beroenden på en högre nivå än en enskild aktör. Ett antal olika delmål är preciserade utifrån huvudmålet och dessa presenteras nedan.

- Beskriva befintliga modeller som kan ligga till grund för utvecklandet av en lämplig beräkningsmetod som svarar mot övriga mål inom detta examensarbete.
- Bestämma om den utvecklade beräkningsmetoden är lämplig för att hantera aggregering av flöden.
- Implementera den utvecklade beräkningsmetoden i ett datorverktyg för att genomföra analyser.
- Att undersöka och identifiera de tilltänkta användarnas behov kopplat till datorverktyget, till exempel:
 - Hur aktörer påverkas av flöden över tiden.
 - Rangordning av de viktigaste flödena, på en aggregerad nivå och för en enskild aktör.
 - Rangordning av de mest kritiska aktörerna, på en aggregerad nivå och för enskilda flöden.
 - Visualisering av utslagning av en eller flera aktörer med avseende på:

- Enskilda aktörer.
- Kategorier av aktörer, till exempel sjukvård eller en viss kommun.
- Utvärdera det framtagna datorverktyget utifrån identifierat behov.

1.4 Målgrupp

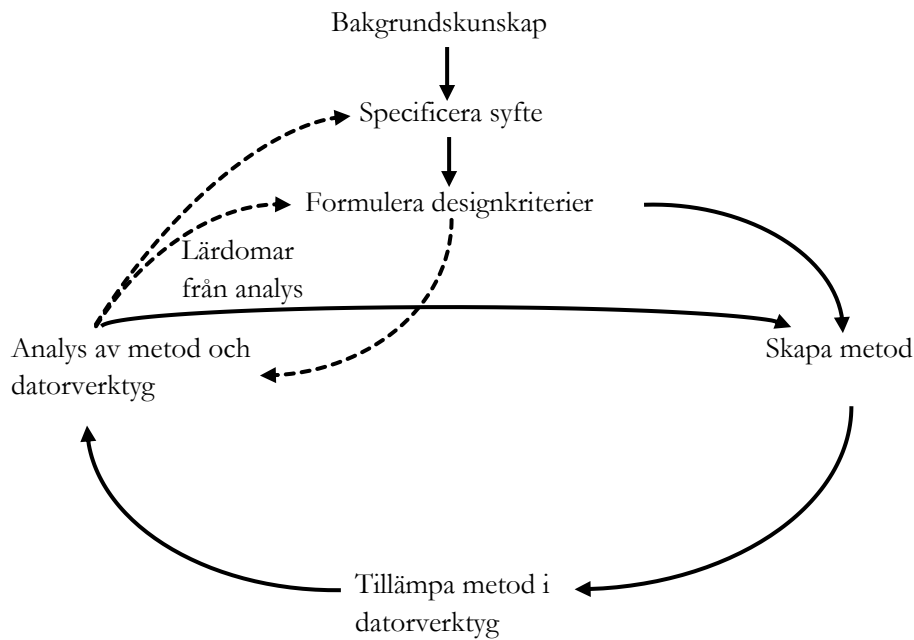
Examensarbetet är avsett för personer som arbetar med kartläggning och analys av beroenden, på kommun- och länsstyrelsenivå men även i privat sektor. Vidare finns det en tydlig koppling mellan detta examensarbete och pågående forskning inom området. Datorverktyget som skapas inom ramen för examensarbetet är endast tänkt att vara en prototyp och det kan vara relevant för andra studenter att vidareutveckla. Prototypverktyget ska i första hand gå att använda för en väl insatt person, även om tanken är att resultatet som kommer från verktyget ska kunna användas direkt. Beräkningsmetoden som ligger till grund för datorverktyget kan vara användbar för personer som är intresserade av att föra arbetet kring beroendeanalys framåt.

1.5 Avgränsningar

Metoder för datainsamling är inte en del i detta examensarbete. De insamlade uppgifterna antags hanteras på ett säkert sätt och därför är inte informationssäkerhet kopplat till datahanteringen en del av rapporten. I kapitel 5.1 beskrivs de antaganden som är gjorda för beräkningsmodellen och datorverktyget, dessa antaganden kan även ses som avgränsningar. Validering av modell och verktyg genomförs inte i större omfattning och därför ska verktygets användas med försiktighet. Verktyget som skapas under examensarbetet är inte tänkt att kunna användas direkt av en oinitierad slutanvändare.

1.6 Plan för genomförande

Projektet har genomförts i form av en *design research*-baserad metodik som presenterats i Hassel (2010). Det finns ett flertal olika metoder som kan vara lämpliga under examensarbetet och målsättningen har varit att använda tillfredställande metoder men inte tvunget de optimala för att begränsa arbetsbelastningen. Arbetsprocessen åskådliggörs i figur 1 och beskrivs mer ingående i etappuppdelningen nedan. Etappuppdelningen är anpassad för projektet och innehåller vissa förändringar och tillägg jämfört med den som presenteras av Hassel (2010).



Figur 1. Översikt över projektets metodik. Inspirerad av Hassel (2010).

1.6.1 Etapp 1 – Bakgrundskunskap

Arbetet inleds med en litteraturstudie för att ge författarna och läsarna en förståelse för kritiska beroenden och flöden, metoder för att hantera detta samt existerande verktyg för att analysera beroenden, se kapitel 2. En djupare analys av existerande modeller redovisas i kapitel 2.6.

1.6.2 Etapp 2 – Specificera syfte

Syftet som definieras i kapitel 1.2 bygger på de behov som identifieras under litteraturstudien. Ändringar i syftet genomförs under projektets gång när det till exempel framkommit önskemål från användare och då lärdomar dras under användningen, se etapp 6 och 7.

1.6.3 Etapp 3 – Designkriterier; identifiera indata och efterfrågade resultat

Etapp 3 består i att identifiera vilken indata som behövs och vilka resultat som efterfrågas, detta styr därefter utvecklingen av analysmetoden. Ett steg för att skapa en bild av området är att intervjua ansvariga för säkerhetssamordning i Lunds kommun. Intervjun ligger till grund för den kravspecifikation som har växt fram. Etappen har lett fram till ett antal mätbara egenskaper och mål som framtagen beräkningsmetod och datorverktyg utvärderas mot. En teoretisk bakgrund till hur intervjuer bör genomföras presenteras i kapitel 3.1 och kravspecifikationen presenteras i Appendix A.

1.6.4 Etapp 4 – Ta fram lämplig beräkningsmetod

Under etappen skapas en beräkningsmetod för analys. Metoder tillämpas på ett antal olika sätt, bland annat i liten eller full skala och på hypotetiska eller verkliga system (Hassel, 2010). I första hand tillämpas beräkningsmetoden på den småskaliga data som samlats in av H. Johansson och Åhsberger (2012) men även på hypotetisk data när datorverktyget verifieras. Framtagandet av antaganden och beräkningsmetod presenteras i kapitel 5.1 och 5.2.

1.6.5 Etapp 5 – Tillämpa beräkningsmetoden och skapa datorverktyg

I etapp 5 programmeras datorverktyget i enlighet med den framtagna beräkningsmetoden. Datorverktyget skapas för att visualisera resultaten så att designkriteriet och mål uppnås. Etappens syfte är att ta fram ett verktyg som uppfyller de ställda kraven. Det framtagna verktyget behöver inte nödvändigtvis vara fullständigt optimerat så länge kraven uppfylls. Koden för datorverktyget presenteras i Appendix D.

1.6.6 Etapp 6 – Analys av metod och verktyg

Detta steg innebär att beräkningsmetoden och resultaten från datorverktyget utvärderas. I kapitel 0 redovisas resultaten och det kontrolleras om de svarar mot kravspecifikationen som definierats under etapp 3. Redan under etapp 5 kommer metoden till viss del analyseras för att därefter leda till förändringar. En del av analysen är att utföra delar av en risk- och sårbarhetsanalys för att undersöka datorverktygets användbarhet, vilket görs i form av en fallstudie i kapitel 7.

1.6.7 Etapp 7 – Lärdomar från analys och användande

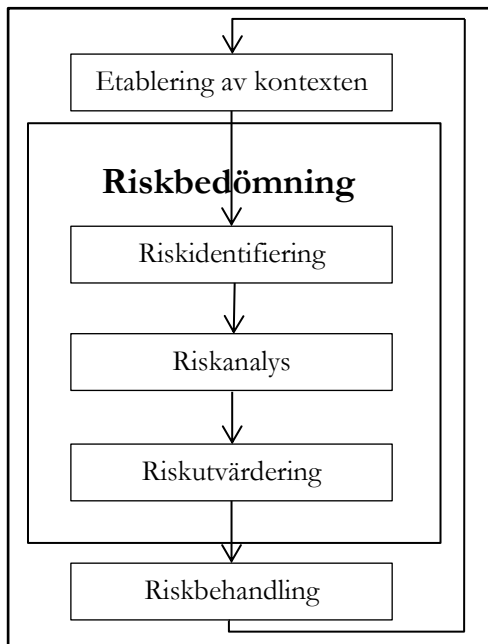
I projektet delas lärdomarna upp i två delar. Den ena delen avser lärdomar kring metoden och den andra delen avser lärdomar direkt inriktade på verktyget. Lärdomarna som är avsedda att ligga till grund för vidare studier inom området och redovisas i kapitel 9.1.

2 Teoretisk bakgrund

Detta kapitel syftar till att ge läsaren en teoretisk bakgrund till ämnet beroendeanalys och fungera som en sammanställning av den litteraturstudie som genomförts. Kapitlet inleds med förklaring av viktiga begrepp och fortsätter därefter med en genomgång av de metoder och datorverktyg som i dagsläget finns tillgängliga. I Kapitel 2.1 och 2.2 presenteras teori kring både riskhantering generellt och risk- och sårbarhetsanalys specifikt. Det som presenteras är avsett att vara en teoretisk introduktion till beroendeanalys ur flödesperspektivet och som hjälp för läsaren att sätta examensarbetet i ett sammanhang och används därefter inte vidare.

2.1 Riskhantering

Riskhantering är den kontinuerliga process som syftar till att identifiera, analysera och behandla en organisations risker (Swedish Standards Institute, 2009). De centrala begreppen i riskhanteringsprocessen beskrivs i standarden SS-ISO 31000:2009 och visas i figur 2. Att riskhanteringsprocessen är kontinuerlig innebär att arbetet aldrig kan anses vara slutfört. Istället bör detta vara en ständigt pågående process och förändringar måste göras när förutsättningarna ändras (ISO, 2009).



Figur 2. Delar av riskhanteringsprocessen, inspirerad av ISO (2009).

MSB (2011b) använder sig av en anpassad version av riskhanteringsprocessen som utöver de steg som redovisas i figur 2 även innehåller sårbarhetsbedömning och sårbarhetsanalys. Processen som MSB (2011b) använder sig av presenteras i figur 3.



Figur 3. Riskhanteringsprocessen beskriven av MSB (2011b). Hämtad från Johansson och Åhsberger (2012, s. 21) med tillstånd.

2.2 Risk- och sårbarhetsanalys

Risk- och sårbarhetsanalys är en viktig del av kommunernas riskhanteringsprocess och uppfyller ett antal syften för kommunen (MSB, 2011b). Ett syfte är beslutsfattare och ansvariga hos kommunen ska få ökad kunskap och medvetenheten kring risker och sårbarheter som återfinns inom kommunens geografiska områdesansvar. Ett annat syfte är att risk- och sårbarhetsanalysen kan användas under samhällsplaneringen och som beslutsunderlag för att motivera investeringsplaner där målet är att öka motståndskraften i samhället. När en kommun genomfört en risk- och sårbarhetsanalys kan den dessutom användas som underlag till samhällets riskbild i stort och som utgångspunkt vid information till allmänheten kring samhällets risker (MSB, 2011b).

I figur 3 syns den riskhanteringsprocess som föreslås i MSB (2011b) och de två moment som används för risk- och sårbarhetsanalys är riskbedömning och sårbarhetsbedömning. Uppdelningen av riskbedömningen visas i figur 3 och de tre delarna är riskidentifiering, riskanalys och riskutvärdering (MSB, 2011b). Riskidentifieringen är den del av riskbedömningen där möjliga händelser, så kallade riskscenarier, identifieras. Riskidentifieringen syftar till att identifiera oönskade händelser och vilka konsekvenser dessa kan ge upphov till. Resultatet från riskidentifieringen användas sedan i riskanalysen där riskscenarierna kvantifieras. Kvantifieringen av scenarierna innebär en uppskattning av hur stor sannolikhet och hur stora konsekvenserna blir om händelsen inträffar. Sista delen av riskbedömningen, riskutvärderingen, är en beskrivning av vilka risker i riskanalysen som kräver vidare åtgärder. Finns det förslag på åtgärder för att minimera riskerna bör dessa inkluderas i riskutvärderingen (MSB, 2011b). Om flera förslag, för samma scenario, föreligger bör dessa värderas med en kostnad-nyttoanalys där även andra för- och nackdelar med förslagen bör lyftas fram.

Sårbarhetsbedömningen delas in i förmågebedömning och sårbarhetsanalys (MSB, 2011b).

Förmågebedömningen beskrivs i MSBFS (2010:6) som en bedömning av förmågan hos samhällsviktig verksamhet att motstå allvarliga störningar och utförs med hjälp av ett antal i förväg angivna indikatorer. Sårbarhetsanalysen är en mer detaljerad form av riskidentifiering som utförs för ett par specifika riskscenarier. Den skiljer sig från riskidentifieringen genom att fokus för sårbarhetsanalysen är på att utförligt beskriva konsekvenserna i riskscenarierna. Sårbarhetsanalysen ska ingående beskriva hur organisationer hanterar, motstår och återhämtar sig efter en händelse (MSB, 2011b).

Kommunerna är enligt MSBFS (2010:6) ålagda att genomföra en risk- och sårbarhetsanalys minst en gång per mandatperiod och följa upp denna årligen. Enligt MSBFS (2010:6) ska en risk- och sårbarhetsanalys behandla följande åtta punkter:

1. *Övergripande beskrivning av kommunen.*
2. *Övergripande beskrivning av arbetsprocess och metod.*
3. *Övergripande beskrivning av identifierad samhällsviktig verksamhet inom kommunens geografiska område.*
4. *Identifierade och värderade risker, sårbarheter samt kritiska beroenden inom kommunens geografiska område.*
5. *Övergripande beskrivning av särskilt viktiga resurser som kommunen kan disponera för att hantera extraordinära händelser.*
6. *Bedömning av förmågan i samhällsviktig verksamhet inom kommunens geografiska område att motstå och hantera identifierade risker som kan leda till en extraordinär händelse.*
7. *Bedömning av kommunens förmåga att motstå och hantera identifierade risker som kan leda till en extraordinär händelse.*

8. *Planerade och genomförda åtgärder samt en bedömning av behov av ytterligare åtgärder med anledning av risk- och sårbarhetsanalysens resultat.*” (MSBFS 2010:6, s. 2-3)

Punkt ett och två kan anses vara en del av utgångspunkterna i den riskhanteringsprocess som beskrivs av MSB (2011). Punkt tre är framförallt en del av riskidentifieringen men kan även innefattas i utgångspunkterna. Värderingen av risker i punkt fyra är en del av riskanalysen och värderingen av sårbarheterna och de kritiska beroendena är en del av sårbarhetsanalysen. Punkt fem till åtta är dels riskvärderingen och dels förmågebedömningen. Som ett verktyg för sårbarhetsanalys rekommenderar MSB (2011) beroendeanalys. För att underlätta kommunernas arbete med att identifiera och värdera beroenden har MSB skrivit handboken *Beroendeanalys – Så här gör du!* (MSB, 2007). Handboken är uppdelad i följande tre delar:

- Urval och beskrivning
- Identifiering och värdering av externa beroenden
- Aggregerad analys

Syftet med datorverktyget som utvecklats är att förbättra det som motsvarar den aggregerade analysen. MSB (2007) beskriver den i två delar, *strukturering av insamlat underlag* och *visualisering av beroendeförhållanden*. Datorverktyget fungerar på liknande vis, först struktureras tillgänglig data och därefter visualiseras resultatet från datorverktyget på lämpligt vis. Framtagandet av beräkningsmetoden som ligger till grund för datorverktygets hantering av data är vidare beskrivet i kapitel 5.2. Visualiseringarna i datorverktyget baseras till viss del på de metoder som beskrivs i MSB (2007). MSB (2007) beskriver hur matriser och beroendekedjor kan användas för att visualisera aktörers beroende av andra aktörer och hur spridningskedjor kan användas för att beskriva hur störningar hos en aktör sprider sig till övriga aktörer. Datorverktyget kan göra motsvarande visualisering och är tänkt att visa på kommunens beroenden samt tydliggöra vilka delar i samhället som är mest sårbara och vilka delar som är viktigast för att samhället ska fungera.

2.3 Flöden

H. Johansson och Åhsberger (2012) beskriver flöden som förflyttningen av varor och tjänster. Dessutom anges underkategorierna människor, energi, kapital eller information. Syftet med dessa sex kategorier av flöden är det ska underlätta identifieringen av enskilda flöden. Flöden kan ibland anses vara svåra att identifiera och definiera eftersom de inte behöver existera fysiskt utan bara representerar en förflyttning. För att ge några exempel på flöden kan följande nämnas; *person- och godstrafik, drivmedel, elektricitet, IT-kommunikation, mobiltelefoni, dricksvatten, avloppsvatten* och *fjärrvärme*. Tanken med flödesbegreppet är att det på ett naturligt sätt ska knyta samman forskningen kring kritisk infrastruktur och samhällsviktig verksamhet. Trots att det kan vara svårt att identifiera flöden möjliggör begreppet en analys av beroenden i samhället som tidigare varit svår att genomföra. Detta görs genom att flödeskonceptet tydliggör tidigare okända beroenden. Aktörer och dess aktiviteter kan påverka eller vara beroende av flöden. För en mer ingående beskrivning av flödesbegreppet rekommenderas Lindström och Wikman (Under tryckning).

2.4 Beroende

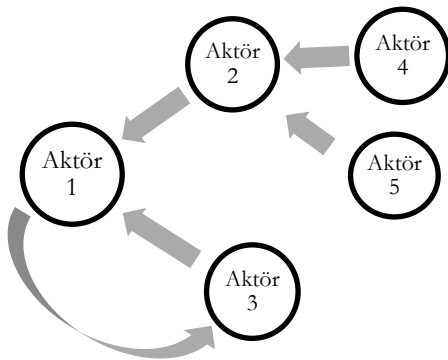
Det finns olika typer av beroende och definitionen varierar med sammanhanget som ordet används i. MSB (2009) definierar beroende som:

”I det här sammanhanget innebär ett beroende en relation mellan två verksamheter, där den ena är beroende av att den andra levererar en vara eller en tjänst.” (MSB, 2009, s. 87)

Flöden är sedan tidigare definierade som förflyttning av varor och tjänster, vilket gör det lämpligt att använda MSB:s tolkning av ordet även i detta examensarbete. Rinaldi et al. (2001) och J. Johansson, Hassel och Cedergren (2011) delar upp beroende i två olika kategorier; enkelriktat och ömsesidigt beroende. Två aktörer är beroende av varandra om en aktör påverkar ett flöde som den andra aktören behöver. I det enkelriktade beroendet påverkar inte den andra aktören den första aktören. Är beroendet däremot ömsesidigt påverkar även den andra aktören den första aktören. Denna påverkan mellan den andra och första aktören benämns ofta som en feedback-loop (System Dynamics Society, 2011).

2.4.1 Indirekt beroende

Aktörer kan ofta vara indirekt beroende av fler aktörer än vad de är direkt beroende av (Rinaldi, Peerenboom, & Kelly, 2001). I figur 4 är aktör 1 beroende av aktör 2 och 3, aktör 3 är beroende av aktör 1 och aktör 2 är i sin tur beroende av aktör 4 och 5. Aktör 1 blir därmed indirekt beroende av aktör 4 och 5. I ett verkligt system är bilden av aktörernas beroende betydligt mer komplex, då beroendekedjorna är längre och flertalet feedback-loopar existerar (Rinaldi, Peerenboom, & Kelly, 2001). Ett exempel på en feedback-loop syns mellan aktör 1 och 3 i figur 4.



Figur 4. Direkta och indirekta beroende mellan olika aktörer.

2.5 Stötdämpare

Stötdämpare hos en aktör har liknande funktion som en stötdämpare i en bil. På samma sätt som bilen skyddas från en hård stöt av ett hinder i vägbanan, skyddar en stötdämpare aktören från en hård stöt vid ett bortfall av ett flöde. Med en stötdämpare kan en verksamhet upprätthålla hela eller delar av sin kapacitet även om aktören påverkats på andra sätt (MSB, 2009). Generellt går det att skilja på interna och externa stötdämpare. Interna stötdämpare finns tillgängliga inom en aktör som till exempel ett eget lager av reservdelar. Externa stötdämpare är istället baserade på andra aktörer. De tre typerna av stötdämpare som kan appliceras i beroendeanalys är redundans, substitut och adaptivitet (MSB, 2009).

Redundans innebär att en resurs som behövs finns tillgänglig i större mängd än vad som normalt används (Little, 2002). Att en aktör har redundans kan till exempel innebära att det finns tillgång till reservkraft om det blir elavbrott eller tillgång till extra personal om den befintliga personalen inte är tillgänglig. Det går att skilja på intern och extern redundans. Ett exempel är ett sjukhus som har intern redundans på tillgång av vatten genom att ha ett eget lager med vatten så att verksamheten kan fortsätta utan det externa flödet av vatten. Extern redundans kan exemplifieras med hur de svenska elnätbolagen tog in personal från Polen och Tyskland för återuppbyggnaden av elnätet efter stormen Gudrun när de egna resurserna inte var tillräckliga (Statens energimyndighet, 2005).

Ett substitut är en resurs som kan ersätta en befintlig resurs vid ett avbrott och därmed upprätthålla tillfredställande kapacitet (MSB, 2009; Zhang & Peeta, 2011). Flödet fast telefoni kan till exempel substitueras med mobil telefoni. Ett annat exempel är hur flexifuelbilar kan använda bensin som substitut för den etanol som vanligtvis används som bränsle till bilen. Om en aktör istället anpassar sig till de förändrade förutsättningarna och upprätthåller viss kapacitet handlar det om adaptivitet. Anpassningen kan till exempel ske genom alternativa produktionsmetoder eller att resurser omfördelas.

2.6 Modeller och verktyg för beroendeanalys

De första stegen i att utveckla ett datorverktyg för att kartlägga och analysera beroende är att undersöka vad som redan utvecklats samt hur kartläggning och analys hanteras idag. Nedan presenteras de modeller och verktyg som av författarna bedömts vara relevanta vid beroendeanalys med hjälp av flöden.

Generellt går det att skilja olika former av modeller som till exempel ekvationsbaserade, analytiska och agentbaserade modeller (Parunak, Savit, & Riolo, 1998). En ekvationsbaserad modell är uppbyggd på ett antal ekvationer och att köra modellen innebär att ekvationerna beräknas. Ett enkelt exempel på detta är en modell med ett antal sammankopplade vattentankar och hur flödet mellan dessa beräknas (Parunak, Savit, & Riolo, 1998). De ekvationer som används i en ekvationsbaserad modell är fördefinierade och kan inte förändras under körningen (Sapkota, 2010). Enligt Sapkota (2010) är en nackdel med ekvationsbaserade modeller, förutom att de inte tar exakt hänsyn till varje aktörs beteende, är att de kräver att allt som ska modelleras definieras med ekvationer.

En agentbaserad modell bygger på att beteendet för varje agent simuleras individuellt. Den metod för beroendeanalys som examensarbetet är baserad på är inte utvecklad för att användning med agentbaserade modeller, utan ekvationsbaserade modeller (J. Johansson, personlig kommunikation, 16 september, 2013). Därmed möjliggör den data som H. Johansson och Åhsberger (2012) sammanställt inte simulering av hur varje aktör agerar internt och agentbaserad modellering är inte lämpligt. På grund av detta återfinns bara en kortare beskrivning av agentbaserade modeller och möjligheterna för beroendeanalys baserat på dessa.

Den huvudsakliga skillnaden mellan agent och ekvationsbaserade modeller är nivån som modelleras. Den ekvationsbaserade modellen undersöker systemet som helhet när det agentbaserade istället simulerar varje enskild agent och resultatet från simuleringen är hur systemet fungerar som helhet. Fördelen med att använda en ekvationsbaserad modell är att det därmed inte krävs data för alla faktorer som kan påverka agenten utan mer systemtäckande ekvationer kan användas.

2.6.1 Nätverksmodeller

Nätverksmodellering är en ekvationsbaserad modell som innebär att system och dess inre kopplingar avbildas och analyseras enbart med hjälp av noder och länkar mellan dessa (J. Johansson, Jönsson, & Johansson, 2007). Noderna är de olika enheterna i systemet och länkarna är kopplingarna mellan dessa. För att beskriva vilka noder som är sammanlänkade och styrkan på dessa kopplingar används en så kallad kontaktmatris i sin enklaste form. En kontaktmatris har storleken $n \times n$ där n är antalet noder och varje element i matrisen är antingen 0 eller 1 för att visa på relationen mellan noderna. Nätverksmodeller lämpar sig väl för analys av teknisk infrastruktur men även för analys av aktörsnätverk (Abrahamsson, 2009). Framförallt eftersom det ofta finns tydliga kopplingar mellan noderna och noderna har tydliga och mätbara funktioner. En nods entydiga funktion gör att det blir lättare att avgöra om den fungerar (1) eller är utslagen (0). Nätverksanalys har främst använts för att modellera enskild teknisk infrastruktur men även beroenden mellan olika former av infrastruktur kan modelleras (Zhang & Peeta, 2011; Johansson J. , 2010).

Traditionell nätverksmodellering är begränsat till att bara modellera system som är homogena genom att alla noder har samma funktion (Johansson J. , 2010). För att bättre kunna ta hänsyn till systemets verkliga

egenskaper och att olika noder har olika funktioner finns vidareutvecklingar. Dessa vidareutvecklingar tillåter enligt J. Johansson (2010) både heterogena system och att specifika noder och länkar har begränsningar. För att använda nätverksmodeller i en sårbarhetsanalys av samhället krävs vissa anpassningar. J. Johansson et al. (2007) inför ett antal nya begrepp och presenterar en metod för att använda nätverksanalys som en del av kommunal eller regional sårbarhetsanalys. Enligt Johansson et al. (2007) är bland annat störningsmått och analysmetoder ofta anpassade för att mäta konsekvenserna på systemet snarare än på samhället som är beroenden av systemet. Genom att kombinera nätverksmodeller med modeller där systemets fysiska egenskaper representeras går det att ytterligare förbättra nätverksmodellerings användbarhet vid sårbarhetsanalys (Johansson J. , 2010).

Även med en nätverksmodell anpassad för sårbarhetsanalys finns det begränsningar i form av vad som kan modelleras. Till exempel bör system utan tydliga noder och länkar vara svåra att hantera. Ett exempel på detta är när det inte finns tillräcklig information för att avgöra hur ett systems fysiska kopplingar till specifika noder ser ut. Tankesättet i nätverksmodellering, med att se system som noder och länkar samt att använda matriser för att hantera kopplingarna mellan dessa, bör däremot utgöra en bra grund för fortsatt utveckling av modeller för att hantera flödesberoenden.

2.6.2 Input-output-modeller

Input-output-modeller är en form av ekvationsbaserade modeller som från början utvecklades av Wassily Leontief år 1951 (Haines & Jiang, 2001). Leontief input-output-modellen utvecklades med syfte att förutse den ekonomiska framtiden. I modellen definieras input som det en sektor behöver för att producera en vara eller tjänst medan output är varan som produceras. Modellen beskriver hur olika sektorer eller industrier i en ekonomi interagerar med hjälp av monetära belopp som förflyttas mellan de olika sektorerna och hur stor efterfrågan på slutprodukten är. Modellen bygger på att varje sektor producerar en vara som andra sektorer är olika starkt beroende av och att sambandet mellan input och output är linjärt. Med hjälp av dessa beroenden kan förändringar i kapaciteten hos en sektor sprida sig till resterande sektorer i samhället.

Haines och Jiang (2001) har anpassat Leontiefs input-output-modell till riskhantering av sammanlänkade ekonomiska sektorer. Anpassningen är gjord genom att både ersätta det monetära belopp som kopplar ihop sektorerna med ett tal som anger grad av inoperabilitet och baserar efterfrågan på slutprodukten på hur stor störningen är. Den engelska termen *inoperability* direktöversätts till svenska och avser systemets oförmåga att utföra sin avsedda funktion (Ginstrup & Rubil, 2013). Grad av inoperabilitet definieras av Haines och Jiang (2001) som en variabel mellan 0 och 1, där 0 innebär att systemet är fullständigt funktionsdugligt och 1 att det är fullständigt utslaget. Input-output-modeller och inoperabilitet har använts för olika former av beroendeanalys utifrån ekonomisk data vid ett antal olika tillfällen (Ginstrup & Rubil, 2013; Haines & Jiang, 2001; Johansson, Svegrupp, & Hassel, 2013; Setola, 2007; Svegrupp, 2012). När modellen användes av Setola (2007) användes expertbedömningar istället för ekonomiska data. Denna anpassning till expertbedömningar är den del av input-output-modeller som ligger närmast för att analysera beroenden genom flöden eftersom workshoppen med respektive aktör kan ses som en form av expertbedömningar. En begränsning med input-output-modeller är att den data som finns tillgänglig framförallt tar hänsyn till hela sektorer eller infrastruktursektorer, vilket gör att resultaten blir begränsade till att gälla på en aggregerad nivå (Johansson, Svegrupp, & Hassel, 2013). Genom att kombinera en regional input-output-modell med en fysik modell för specifik infrastruktur argumenterar Johansson et al. (2013) för att det går att hämta de bästa egenskaperna hos de båda modelleringsformerna och därmed visa konsekvenserna för samhället i stort vid störningar i specifik infrastruktur.

Ytterligare en begränsning med input-output-modeller är att sambandet mellan input och output förenklats till att vara linjärt (Zhang & Peeta, 2011). I fall där det krävs en enhet input för att en aktör ska skapa en enhet output kan detta antagande vara korrekt, men Zhang och Peeta (2011) menar att det

generellt inte är ett realistiskt antagande i en komplex verklighet. Dessutom är inte input-output-modeller i grunden avsedda att ta hänsyn till förändringar över tiden. Zang och Peeta (2011) menar att allt i systemet kommer bli otillgängligt om beroenden tillåts fortplanta sig obehindrat. Haimes och Jiang (2001) har modifierat modellen för att möjliggöra analyser med tidssteg men även med denna modifiering blir hela systemet otillgängligt när antalet tidssteg går mot oändligheten. Trots dessa begränsningar anser Setola (2007) att modellen lämpar sig väl för att på ett enkelt sätt beskriva beroenden och oberoenden i komplexa system. Detta eftersom den ekonomiska informationen som krävs ofta finns tillgänglig på nationell nivå (Svegrupp, 2012). Att informationen framförallt finns tillgänglig på nationell nivå begränsar exaktheten i användningsmöjligheterna för input-output-modell. När det finns tillgång på information menar Zhang och Peeta (2011) att input-output-modellen har en stor fördel genom att den möjliggör matematisk analys av beroende och risköverföring av ömsesidigt beroende infrastrukturektorer.

2.6.3 System dynamics

System dynamics är inte en metod eller verktyg på samma sätt som nätverks- och Leontiefs input-output-modellen. Istället är det ett tillvägagångs- eller tankesätt för att hantera system med delar som ständigt påverkar varandra (System Dynamics Society, 2011). Påverkan sker genom så kallade feedback-loopar från andra delar av systemet. Hanteringen av feedback är en av grundstenarna inom system dynamics och en central del av detta är cirkulära orsakssamband (System Dynamics Society, 2011). Begreppet cirkulära orsakssamband används för att beskriva hur en aktör indirekt kan påverka sig själv. Ett exempel på detta är när aktör 1 som både påverkar och är beroende av aktör 3 får en störning och därmed påverkas ytterligare av aktör 3, se figur 4. System dynamics är från början utvecklat för att användas inom industrin, men har numera fått stor spridning, till exempel används det både för konfliktförebyggande i skolmiljö och miljöanalyser (Ahlin, Ardefel, & Sundström, 2005; System Dynamics Society, 2011). System dynamics hanterar feedback-loopar genom att förändringar beräknas med första ordningens differentialekvationer och spridningsberäkningarna sker i flera tidssteg.

De beroendeanalyser som idag utförs är baserade på riktlinjer från MSB (2007) och utvärderar ofta bara beroenden i ett steg och därmed tillämpas inte system dynamics fullständigt. För att en beroendeanalys ska kunna hantera indirekta beroenden bör tankesätt från system dynamics tillämpas i en större utsträckning. Bland annat genom att ta hänsyn till feedback-loopar och cirkulära orsakssamband.

2.6.4 Agentbaserade modeller

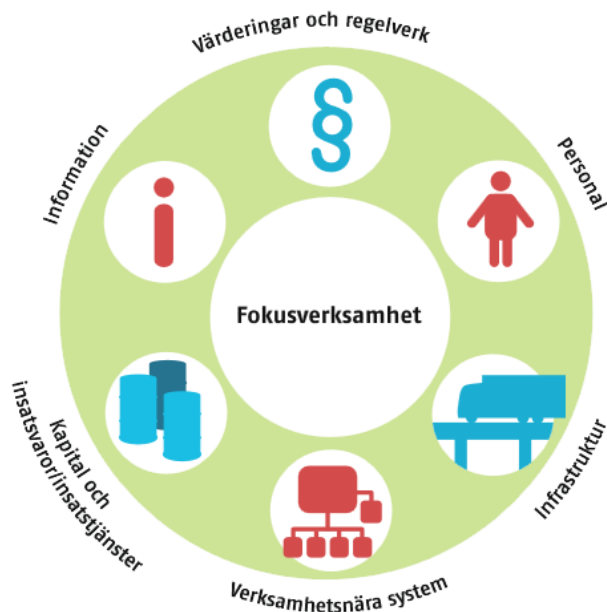
Agentbaserad modellering är en modelleringsform som ökat i popularitet i och med att datorkapaciteten utvecklats och intresset ökat för att förstå komplexa system som till exempel system av celler i levande organismer (Macal & North, 2006). Modelleringen bygger på att varje nod eller agent i systemet kan fatta egna beslut och att varje beslut är representerat i modellen och därmed kan påverka resten av modellen. En enskild cell i en varelse kan ses som en agent, precis som en enskild aktör kan i det här examensarbetet. Modelleringen bygger på att varje agents beteende och beslut simuleras fullständigt och därmed krävs både regler och indata som tillåter att agenten fattar oberoende beslut. När system uppbyggda av flera agenter simuleras kan olika agents påverkan på varandra och resultaten för hela systemet observeras. Varje agent är uppbyggd kring en funktion med ett minne och ett objekt med förmåga att agera autonomt (Macal & North, 2006). En agentbaserad modell har därmed ingen central styrning som kan sätta regler som påverkar olika funktioner, modellen eller hur systemet går mellan olika lägen.

Det finns olika agentbaserade modeller som är utvecklade för att hantera ömsesidiga beroenden hos infrastrukturer, se till exempel North (2001) samt Dudenhoefter, Permann och Manic (2006). Modellen utvecklad av Dudenhoefter et al. (2006) är dessutom kommersialiserad i form av programvaran TACCS och används idag framförallt för att ge en överblick vid inträffade händelser och för att prioritera insatser (Idaho National Laboratory, u.d.). Programvaran ska till viss del kunna skapa en egen bild av

samhällssystemet genom att samla in finansiell information autonomt (Zhang & Peeta, 2011). Bilden bör dock kompletteras eftersom alla former av beroenden inte åskådliggörs med finansiella kopplingar (Setola, 2007). Information kan matas in manuell i TACCS med hjälp av en karta eller ett flygfoto där olika kritiska samhällsfunktioner kan ritas in. Denna enkla metod för inmatning ska göra programmet mer lättanvänt och lätt att implementera i en organisation (Idaho National Laboratory, u.d.). Samtidigt medför den förenklade inmatningen att även analysen är baserad på samma överblicksbild och därmed är det inte möjligt att genomföra en djupare matematisk analys av resultaten. Ytterligare en begränsning med TACCS, men även andra agentbaserade modeller, är mängden data som måste samlas in för att åstadkomma ett tillförlitligt resultat.

2.6.5 Manuella undersökningsmetoder och kvalitativa rapporter

Att beskriva beroenden kvalitativt är en grundläggande metod för analys som ofta är tillräcklig för att överblicka mindre system eller öka medvetenheten kring beroenden (MSB, 2011b). Beskrivande beroendeanalys används idag för kommunala risk- och sårbarhetsanalyser, se till exempel Lunds kommun (2012). Det finns hjälpmedel och metoder utvecklade för att underlätta kvalitativa beroendeanalys. En av dessa är Beroendehjulet som utvecklades av MSB för att fungera som metodstöd vid beroendeanalys (MSB, 2011c). Verktyget hjälper aktören att identifiera och värdera beroenden utifrån de sex kategorier som återges i figur 5. Utöver Beroendehjulet finns det dessutom stöd för aggregering framtaget i form av en handbok (MSB, 2007). Nackdelen med metoden är det är att svårt att åskådliggöra beroenden för komplexa system. Möjligheterna till ytterligare analyser som *what-if* och aggregering är dessutom begränsade eftersom detta måste göras manuellt.



Figur 5. Beroendehjul som kan användas för att identifiera beroenden. Från MSB (2011c).

2.6.6 Supply Chain Risk Management

Supply Chain Risk Management är gränslandet mellan supply chain management och risk management. Supply chain management används för att hantera flödet av produkter från underleverantörer till slutkund (Paulsson, 2007). Risk management handlar till viss del om att identifiera och övervaka risker som finns. Supply Chain Risk Management handlar således om att hantera de risker som finns i ett produktflöde. Att undersöka de verktyg som finns för Supply Chain Risk Management bör vara intressanta eftersom

liknande metoder som används för att hantera en distributionskedja kan användas för att analysera flödesberoenden.

2.6.7 Summering av tillgängliga modelleringsalternativ

De flesta av de metoder och modeller som presenteras i kapitel 2.6 är fortfarande i utvecklingsstadiet även om vissa används i olika sammanhang. Generellt kräver samtliga presenterade metoder, förutom kvalitativa rapporter, stora mängder data. Den data som samlats in för beroendeanalys ur ett flödesperspektiv har inte den detaljnivå som krävs för till exempel agentbaserad modellering. Samtidigt medger beroendeanalys ur ett flödesperspektiv analys med högre detaljnivå än till exempel input-output-modeller. Avvägningen mellan hög detaljnivå på resultatet och förhållandevis enkel datainsamling kan anses vara en fördel med beroendeanalys ur flödesperspektivet. Kvalitativa rapporter och manuella metoder går att använda med begränsad indata men dessa begränsas istället av komplexa system med flera aktörer och flöden. Komplexiteten medför att det blir för svårt att analysera systemet manuellt. Med manuella analyser bör dessutom mycket information gå förlorad vid aggregering eftersom det är svårt att manuellt ta hänsyn till all data på en högre nivå. Tankesättet från system dynamics är lämpligt för att hantera kedjor av indirekta beroenden och kommer att användas i det datorverktyg som utvecklas som en del av detta examensarbete. Supply Chain Risk Management har visat sig vara olämpligt att applicera då det är svårt att hitta detaljerad information om beräkningsmetoderna kopplade till Supply Chain Risk Management.

Ingen av de metoder som presenterats i kapitel 2.6 är väl lämpad för att hantera den data som ligger till grund för detta examensarbete där samtliga aktörer definierats med både styrka på beroende och påverkan av flöden. Därmed måste tillgängliga beräkningsmetoder modifieras eller en ny beräkningsmetod utvecklas för möjliggöra beroendeanalys ur ett flödesperspektiv. Att utveckla en ny beräkningsmetod ligger närmast till hands eftersom det då upplevs som enklare att ta hänsyn till målen och kraven. Genom att skapa en egenutvecklad beräkningsmetod går det att inspireras av fördelarna från tillgängliga modeller samtidigt som nackdelarna kan minimeras. Med en egenutvecklad beräkningsmetod går det dessutom att ta hänsyn till påverkan som sänker en aktörs kapacitet utan att slå ut den vilket till exempel inte en ren nätverksmodell klarar av. Samtidigt behöver inte hela systemet kvantifieras på samma nivå som krävs för input-output-modeller eller agentbaserade modeller där enskilda aktörers beteende måste modelleras.

3 Metod

I följande kapitel presenteras de specifika metoder som används för att utföra moment som intervjuer och utveckling och verifiering av datorverktyg.

3.1 Metod för användarintervju

När kravspecifikationen för datorverktyget skapas är det viktigt att den tar hänsyn till potentiella användares önskemål. För att fånga upp de önskemål som finns kring både resultat och möjlighet till olika former av indata används användarintervjuer. Det finns många olika former av intervjuer, Höst, Regnell, och Runeson (2006) nämner bland annat *öppet riktad*, *halvstrukturerad* och *strukturerad* som olika typer av intervjuer.

En *öppet riktad* intervju är en intervju där personen som leder intervjun har kategorier med frågor. Frågorna är konstruerade så att flera svar är möjliga, vilket ger personen som intervjuas möjlighet att uttrycka sig fritt. Då flera svar är möjliga kan den intervjuade till viss del påverka i vilken omfattning som de olika områdena behandlas. *Strukturerade* intervjuer ger däremot inte den intervjuade möjlighet att styra intervjun och kan närmast liknas vid en enkät. Fördelen mot en enkät enligt Höst et al. (2006) är att det under en *strukturerad* intervju finns det utrymme att förtydliga eventuella oklarheter i frågor och svar. Det finns även en *halvstrukturerad* intervju där vissa delar av intervjun är mer av *öppet riktad* karaktär och andra mer av *strukturerad* karaktär. Under en *halvstrukturerad* intervju säkerställs att specifika frågor får ett koncist svar samt att det finns utrymme för intervjupersonen att lämna övrig information.

Då ett av examensarbetets mål är att uppfylla användarnas önskemål i så hög utsträckning som möjligt har en *halvstrukturerad* intervju genomförts vid mötet med de tilltänka användarna. Målet med mötet var att skapa en enklare kravspecifikation. Med enklare kravspecifikation menas i det här fallet att användarna får sätta upp önskemål för vilken utdata som datorverktyget ska innehålla och hur den ska presenteras. Kravspecifikationen som skapats utifrån intervjuerna presenteras i Appendix A.

3.2 Metod för utveckling av datorverktyg

Datorverktyget utvecklades med hjälp av programvaran MATLAB och baseras på en egenutvecklad beräkningsmetod. Utvecklingen av datorverktyget inleddes med skapandet av den funktion som beräknar spridning av störningar och hur påverkan aggregeras hos flöden och aktörer. Därefter skapades specifika funktioner för att visualisera data eller enskilda analyser som uppfyller den kravspecifikation som skapades under användarintervjun, se Appendix A. Datorverktyget skapades för att vara modulärt och separata funktionsfiler används i största möjliga utsträckning när samma funktion efterfrågas av olika delar av verktyget. De separata funktionerna och deras krav på input och output dokumenterades i ett flödesdiagram som redovisas i Appendix B. Den generella utvecklingen dokumenterades och specifika lösningar och antaganden presenteras i kapitel 5.1. Verifiering och i viss mån validering av datorverktyget presenteras i kapitel 6.3. Datorverktygets kod återfinns i Appendix D.

3.3 Validering och verifiering

För att kunna säkerställa det utvecklade verktygets lämplighet används verifiering och validering. Nedan presenteras tillvägagångssätt för validering och därefter verifiering. Validering är processen som säkerställer att modellen som är implementerad i datorverktyget är en lämplig beskrivning av de aspekter hos det verkliga systemet som ska modelleras (Department of Defense, 2007). Att validera att en modell är en lämplig representation av systemet som undersöks är ofta en svår uppgift (Law & Kelton, 2000). Möjligheterna till validering kan både begränsas av tillgången på tid och resurser, men också av komplexiteten i det modellerade systemet. Modellen som skapas bör valideras stegvis under utvecklingen i

den omfattning som det är möjligt. Det är viktigt att validera modellen specifikt för de scenarier den är avsedd att användas för (Law & Kelton, 2000).

För att genomföra en validering finns ett flertal olika metoder (Kosterev, Taylor, & Mittelstadt, 1999; Law & Kelton, 2000; Sargent, 2005). Ett urval av dessa som kan vara lämpliga för validering av den utvecklade modellen presenteras nedan. Sargent (2005) blandar till viss del ihop validerings- och verifieringsmetoder och i dessa fall presenteras metoderna endast under verifiering baserat på en uppdelning från Law och Kelton (2000). H. Johansson och Åhsberger (2012) har utfört viss validering av flödesmodellen genom att låta varje aktör granska den indata som sammanställts. De metoder som anses vara relevanta för validering är:

- Validera modellen med expertbedömningar. Detta innebär att personer med kunskap kring det modellerade systemet får bedöma om modellens resultat verkar rimligt vid en kortare anblick (Sargent, 2005). Detta kan bara användas i begränsad omfattning i examensarbetet eftersom systemet som modelleras är komplext med långa kedjor av beroenden. Dessa långa kedjor av beroenden är också anledningen till att det finns behov av ett datorverktyg.
- Validera med andra modeller och jämför utdata från modellen med utdata från andra validerade modeller (Sargent, 2005). Detta bör vara en lämplig valideringsmetod, till exempel genom att använda en input-output-modell på samma sätt som Setola och De Porcellinis (2008) och Haimes och Jiang (2001). Arbetsinsatsen som krävs för att bygga upp samma testfall i två olika modeller ligger dock utanför ramarna för detta examensarbete.
- Genomföra enkla testfall där resultaten är förutsägbara även i ett komplext system (Sargent, 2005). Till exempel ett system med ett flertal olika vattentankar som har olika in- och utflöden där det med rätt val av flöden bör gå att förutsäga nivån i en viss tank.
- Göra extremvärdestest, till exempel genom att ansätta kapaciteter på 0 % eller på 100 % och kontrollera om resultaten blir som väntat (Sargent, 2005). Metoden kan i första hand anses vara avsedd för verifiering men även lämpa sig väl för att kontrollera att det modellerade systemet är korrekt återgivet. Metoden sammanfaller delvis med föregående punkt som handlar om enkla testfall.
- Validera baserat på historisk data och förutsägelser om kommande utveckling (Sargent, 2005). Detta kan göras till exempel genom simulering av ett scenario som hänt i verkligheten och kontroll av hur väl resultaten överensstämmer med modellens. Kosterev et al. (1999) har till exempel validerat elnätsmodeller med hjälp av data från stora elavbrott. En metod liknande att validera mot historisk data kan vara att förutsäga framtiden och undersöka om förutsägelseerna stämmer med de verkliga resultaten. Att validera mot historisk data eller förutsägelser av framtiden kan vara en lämplig metod för att validera den i examensarbetet framtagna modellen, men arbetsinsatsen som krävs är större än vad som ligger inom ramarna för detta examensarbete.
- Jämföra resultaten från samma utslagning i ett enkelt testsystem och i systemet med verklig data. Om det går att observera samma tendenser och eventuella avvikelser går att förklara, kan metoden användas för att styrka modellens validitet. Jämförelsen är förhållandevis enkel att utföra och bör lämpa sig väl för en enkel form av validering av den framtagna modellen (Hassel, personlig kommunikation, 16 september, 2013).

Modellen som tas fram i examensarbetet kommer endast att valideras i begränsad omfattning. Validering kommer i första hand ske i form av enkla testfall och extremvärdestest. Datorverktyget blir därmed begränsat till att framförallt vara giltigt för dessa. I viss mån kan expertbedömningar användas, men möjligheterna till detta är begränsade eftersom det är svårt för en expert att ha en korrekt och övergripande bild av det komplexa system som modelleras. Dessa valideringsmetoder är inte tillräckliga för att säkerställa modellens lämplighet för att hantera verkliga system men kan ändå vara ett sätt att

upptäcka vissa brister. Övriga valideringsformer presenteras som en bakgrund till valet av valideringsmetod och för att visa vad som krävs för att den framtagna modellen ska anses vara validerad.

Verifiering är processen där det kontrolleras att verktyget uppfyller konceptbeskrivning, specifikationer och mål som definierats tidigare (Department of Defense, 2007). Detta kan ses som en kontroll av att datorverktyget beräknar det som angetts i konceptbeskrivning och att beräkningarna utförs korrekt. Law och Kelton (2000) beskriver ett antal olika metoder för att verifiera simuleringsverktyg. Ett urval av dessa som anses vara lämpliga för att verifiera det utvecklade verktyget presenteras nedan.

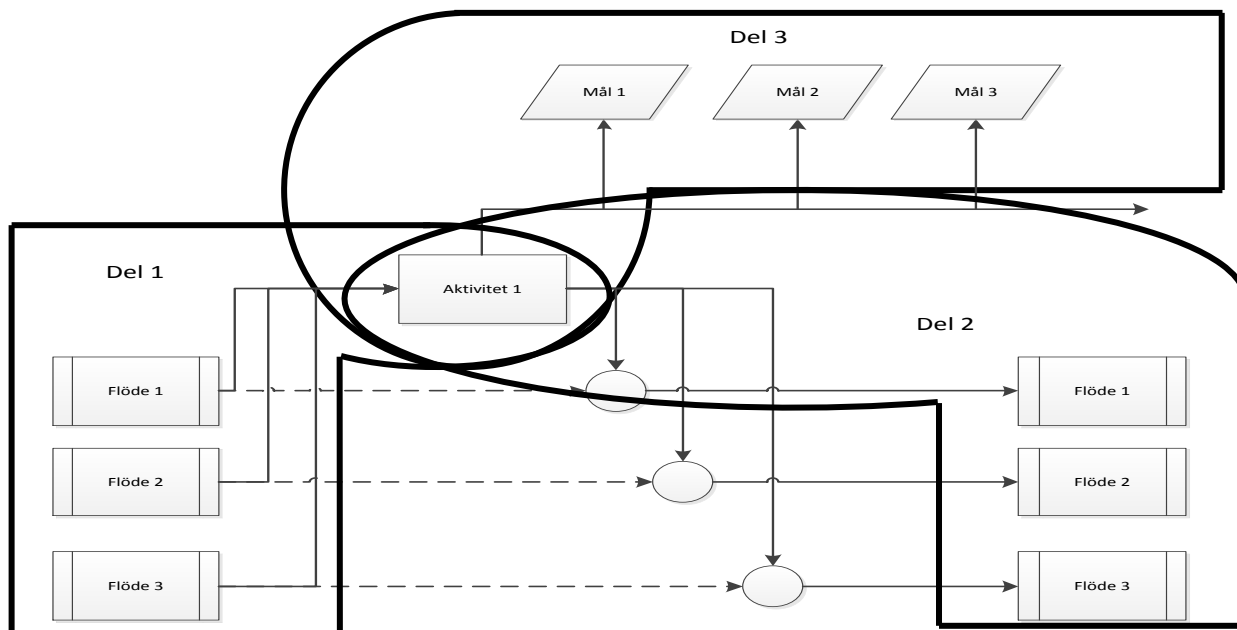
- Utvärdera verktygets huvudfunktion och subfunktioner var för sig under utvecklingens gång. I de fall då specifik data krävs för subfunktioner och denna data inte är tillgänglig kan det hanteras med hjälp av fabricerad data (Law & Kelton, 2000).
- Tillämpa strukturerade genomgångar av programkoden och dess funktioner. Strukturerad genomgång innebär att koden granskas tillsammans av projektdeltagarna och varje sats diskuteras så alla är överens om innebörd och funktion (Law & Kelton, 2000).
- Kör ett antal olika testfall och kontrollera om utfallen blir som förväntat. En variant av detta är att köra enkla testfall där resultaten kan jämföras med handberäkningar (Law & Kelton, 2000). Metoden har likheter med en metod för validering som finns beskriven ovan men kan anses vara relevant även för verifiering.
- Kör koden i spårnings- eller debug-läge, resultaten från varje sats redovisas och jämförs med handberäkningar om det behövs (Law & Kelton, 2000).

Samtliga verifieringsmetoder som presenteras ovan kommer att användas i olika delar av arbetet.

Verifieringen bör vara tillräcklig för att säkerställa att datorverktyget behandlar den framtagna modellen korrekt. Den verifiering som utförts i form av testfall presenteras i kapitel 6.3.

4 Empirisk data

Den data kring beroenden som finns tillgänglig är uppbyggd kring aktörer och vilka aktiviteter som respektive aktör genomför. För att koppla ihop aktörerna används deras aktiviteter och deras påverkan på och beroende av flöden. Samtlig data som används är uppbyggd på samma sätt, se figur 6, och strukturen bygger på den metodutveckling J. Johansson och Hassel utfört (J. Johansson personlig kommunikation, 16 september, 2013).



Figur 6. Bild över hur insamlad data är uppbyggd.

Datansamlingen bygger på en metodutveckling som J. Johansson och Hassel utfört (J. Johansson personlig kommunikation, 16 september, 2013). Den utvecklade metodiken utgår från aktörens viktigaste mål, del 3 i figur 6, och fortsätter därefter med vilka aktiviteter som krävs för att uppnå dessa mål. Aktiviteternas betydelse bedöms med hjälp av alternativen i tabell 1. Aktiviteterna beskrivs sedan i två delar, först vad respektive aktivitet har för flödespåverkan, tabell 2, och vad respektive aktivitet har för flödesberoende, tabell 3. Hur flödesberoendet påverkar aktiviteterna och aktiviteternas flödespåverkan bedöms utifrån sju olika tidsperspektiv. De valda tidsperioderna är: 0-2 timmar, 2-8 timmar, 8-24 timmar, dygn, vecka, månad, år.

Tabell 1. Aktiviteternas betydelse för aktörens mål, del 3 i figur 6. Från H. Johansson och Åhsberger (2012, s. 38) med tillstånd.

1	Aktiviteten har ingen betydelse för om målet uppnås eller inte.
2	Aktiviteten har liten betydelse för om målet uppnås eller inte
3	Aktiviteten har ganska stor betydelse för att målet ska kunna uppnås
4	Aktiviteten är av stor betydelse för att målet ska kunna uppnås
5	Aktiviteten är direkt avgörande för om målet uppnås eller inte

Tabell 2. Aktiviteternas flödespåverkan, del 2 i figur 6. Från H. Johansson och Åhsberger (2012, s. 38) med tillstånd.

0 %	Påverkas inte i någon utsträckning (flödet påverkas inte alls om aktiviteten inte kan utföras)
1-25 %	Påverkas i liten utsträckning (flödet påverkas endast marginellt om aktiviteten inte kan utföras)
26-50 %	Påverkas i medelstor utsträckning (flödet kan upprätthållas men med vissa begränsningar)
51-75 %	Påverkas i stor utsträckning (endast en mindre del av flödet kan upprätthållas)

76-99 %	Påverkas i mycket stor utsträckning (flödet får mycket svårt att upprätthållas)
100 %	Påverkas i full utsträckning (flödet kan inte upprätthållas överhuvudtaget)

Tabell 3. Aktiviteternas flödesberoende, del 1 i figur 6. Från H. Johansson och Åhsberger (2012, s. 38) med tillstånd.

0 %	Påverkas inte i någon utsträckning (aktiviteten påverkas inte alls om flödet inte finns tillgängligt)
1-25 %	Påverkas i liten utsträckning (aktiviteten påverkas endast marginellt om flödet inte finns tillgängligt)
26-50 %	Påverkas i medelstor utsträckning (aktiviteten kan upprätthållas men med vissa begränsningar)
51-75 %	Påverkas i stor utsträckning (endast en mindre del av aktiviteten kan upprätthållas)
76-99 %	Påverkas i mycket stor utsträckning (aktiviteten får mycket svårt att upprätthållas)
100 %	Påverkas i full utsträckning (aktiviteten kan inte upprätthållas överhuvudtaget)

Datainsamling har skett vid tre olika tillfällen och har berört aktörer verksamma i Malmö och/eller Lunds kommun. Insamlingen påbörjades av H. Johansson och Åhsberger (2012) samtidigt som de jämförde olika insamlingsmetoder och aktörerna valdes för att de var intresserade att bidra till insamlingsmetodens utveckling. Data samlades in för följande offentliga aktörer:

- Miljöförvaltningen
- Kultur och fritidsförvaltningen
- Barn och skolförvaltningen
- Renhållningsverket
- Gatukontoret

Under våren 2013 följdes H. Johansson och Åhsbergers (2012) datainsamling upp med tre privata aktörer. Datainsamlingen genomfördes av Jonas Johansson, LTH, och Kajsa Helmbring, tidigare Training Regions(J. Johansson personlig kommunikation, 16 september, 2013), och metoden som användes var workshop i enlighet med H. Johansson och Åhsberger (2012). Data samlades in för följande privata aktörer:

- Copenhagen Malmö Port (CMP)
- EON
- Sveriges åkeriföretag

Den sista datainsamlingen utfördes parallellt med detta examensarbete under hösten 2013. Insamlingen har skett enligt samma metodik som presenterats av med H. Johansson och Åhsberger (2012) och beskrivs mer ingående i Lindström och Wikman (Under tryckning). Datainsamling var planerat för följande aktörer:

- Räddningstjänsten Syd
- Skånetrafiken
- Skånes Universitetssjukhus

Eftersom examensarbetet av Lindström och Wikman utförts under samma tidperiod som detta examensarbete har inte all indata funnits tillgänglig i tid för analysen inom ramen för detta examensarbete.

Det är med hjälp av data från de åtta aktörerna som samlats in innan hösten 2013 som datorverktyget har skapats. För fallstudien av Lunds kommun används även indata insamlad av Lindström och Wikman (Under tryckning) och då enbart för Räddningstjänsten Syd då övrig indata inte funnits tillgänglig i tid. Datorverktyget har däremot utvecklats för att kunna hantera ett obegränsat antal aktörer.

5 Utveckling av beräkningsmetod och datorverktyg

Utveckling av beräkningsmetod och datorverktyg följer etapp 5, kapitel 1.6.5, och den metodik som presenteras kort i kapitel 3.2. I följande kapitel presenteras antaganden och val relaterat till det datorverktyg som utvecklas för att analysera tillgänglig empirisk data som presenteras i kapitel 4.

5.1 Antagande

För att skapa en beräkningsmetod krävs ett antal olika antaganden. Det finns ett flertal antaganden som krävs för att skapa en modell av flödesberoenden. Delar av dessa kan härledas direkt till tidigare arbete kring flöden och andra till utvecklingen av datorverktyget. Antaganden kan delas upp i sådana som är av generell karaktär och sådana som är direkt kopplade till tillgänglig indata. Samtliga antaganden presenteras nedan tillsammans med relevanta följdantaganden.

5.1.1 Geografiska antaganden

De antaganden som gjorts kring aktörers och flödens kopplingar till olika geografiska områden presenteras nedan.

- Aktörers påverkan och beroenden antas gälla inom den egna kommunen eftersom geografisk information kring detta saknas. Den enda geografiska information som finns tillgänglig är den geografiska plats som respektive aktör i huvudsak är verksam på. Däremot saknas data kring om geografisk påverkan och flödernas geografiska utsträckning. Till exempel kan en aktör i Lund påverka godstransporter i Malmö, och därmed flödet *transporter* i Malmö. Så som modellen hanterar tillgänglig data, påverkar aktören i Lund bara flödet *transporter* i Lund. Modellen saknar alltså möjlighet att visa hur aktören i Lund även påverkar aktörer och flöden i Malmö.
- Aktörernas geografiska placering i tillgänglig data är uppdelad på två kommuner, Malmö och Lund. Analysen med datorverktyget kommer att utföras i form av en fallstudie för data insamlad för Lunds kommun men för att kontrollera datorverktygets möjligheter att hantera ett större antal aktörer kan alla aktörer, felaktigt, placeras i samma geografiska område i kapitel 0.

Med de antaganden som presenterats ovan kan inte en aktör påverka eller vara beroenden av aktörer och deras koppling till flöden i annat än den egna kommunen. Detta antagande kan anses vara orealistiskt men krävs för att insamlad data ska vara användbar i examensarbetet. För att utveckla metodiken kring flöden ytterligare föreslås en förbättring i hur data ska samlas kring flödernas geografiska koppling. Detta förslag presenteras i kapitel 8.3.2.

5.1.2 Analytiska antaganden

De antaganden som är direkt kopplade till beräkningarna som datorverktyget utför presenteras nedan. För att underlätta förståelsen av antagandena kan det förtydligas att en aktör är uppbyggd av ett antal aktiviteter som tillsammans avgör dess förmåga att uppfylla angivna mål. Detta beskrivs mer ingående i kapitel 4.

- Påverkan på flöden av aktiviteter och aktivitetens beroende av flöden beräknas på samma sätt. Detta antagande utvecklas vidare i kapitel 5.2.2.
- När den största förändringen i kapacitet mellan två beräkningsiterationer är mindre än 1 % avbryts simuleringen av störningsspridning och systemets anses vara stabilt. En iteration innefattar alla beräkningar som görs från det att aktiviteternas beroende av flödena beräknats en gång och till dess att aktiviteternas beroende av flödena återigen ska beräknas. Om systemet inte blivit stabilt efter 1000

iterationer avbryts simuleringen och ett meddelande ges på skärmen. Båda villkoren går att ändra om simuleringen tar orimligt lång tid med större system.

- Aktörers kapacitet baseras på ett viktat medelvärde av kapaciteten hos samtliga av aktörens aktiviteter. Viktningen är baserad på hur viktig respektive aktivitet är för att uppfylla aktörens mål. I modellen antas att målen, inbördes, är lika viktiga för aktören eftersom det inte finns någon data tillgänglig huruvida målen är olika viktiga. Aktiviteternas viktighet för varje mål har genom ranking getts 1-5 poäng i enighet med tabell 1. För att exemplifiera kan en aktör som utför två aktiviteter användas. Om en aktivitet är fullständigt utslagen och den andra har full kapacitet kommer aktören ha kapacitet kvar. Den kvarvarande kapaciteten baseras på hur stor del av aktörens måluppfyllnad som den fungerande aktiviteten står för. I tabell 4 illustreras en aktör med två aktiviteter, A_x , och tre mål, M_y . För att beräkna aktörens viktade kapacitet, K , adderas poängen från ranking av varje aktivitet samman och divideras med den totala summan av alla aktörens aktiviteter.

Tabell 4. Matris för en aktör som visar exempel på aktiviteternas kapacitet och betydelse för att uppnå aktörens mål.

Aktivitet	Mål			Aktörens beroende	Kapacitet per Aktivitet
	M_1	M_2	M_3		K_A
	A_1	4	5	3	67 %
A_2	2	2	2	33 %	80 %

A_1 i tabell 4 har 12 poäng och A_2 har 6 poäng. Antalet poäng divideras sedan med det totala antalet poäng för hela aktören, vilket för aktören i tabell 4 är 18 poäng. Detta innebär att aktören är 67 % beroende av aktivitet A_1 och 33 % beroende av A_2 . Detta innebär i sin tur att om aktiviteternas kapacitet är 60 % för A_1 och 80 % för A_2 kommer aktörens kapacitet att vara:

$$0,67 \cdot 0,6 + 0,33 \cdot 0,8 = 0,67$$

Aktörens kapacitet influeras alltså mest av aktivitet A_1 . Alternativet skulle kunna vara att anta att måluppfyllandet inte spelar någon roll för aktörens kapacitet. Beräknas kapaciteten utan att ta hänsyn till aktörens mål skulle den få följande kapacitet:

$$0,5 \cdot 0,6 + 0,5 \cdot 0,8 = 0,70$$

Målen indikerar vilken av aktiviteterna som är viktigast för aktören och detta bör avspeglas i hur aktiviteternas kapacitet påverkar aktörens kapacitet.

5.1.3 Antaganden kopplade till tillgänglig indata

Nedan presenteras de antaganden som är kopplade till hur tillgänglig indata tolkas.

- Medelvärdet i de spridningsintervall som aktörerna angett för påverkan och beroende har använts. Intervallen presenterades mer ingående i kapitel 4. För att genomföra känslighetsanalyser och visa på spridningen i resultaten används även minsta och högsta värden i vissa av datorverktygets funktioner.
- Fullständig linjäritet mellan kapacitet och möjlighet till påverkan antas. Det vill säga att om en aktivitets kapacitet är nedsatt kommer aktivitetens möjlighet att påverka flöden sänkas i motsvarande grad. Ett exempel är en aktivitet som kan påverka ett flöde med 100 %. Om aktivitetens kapacitet är nedsatt med 60 % kommer flödets kapacitet att sänkas med 60 %. Någon korrelation mellan beroendet och påverkan finns inte angivet i H. Johansson och Åhsberger (2012). I brist på indata för att kunna undersöka andra alternativ, antas linjäritet. För att bedöma om linjäritet är ett lämpligt antagande krävs ytterligare information där aktörerna får bedöma hur olika kapaciteter för aktiviteter är kopplade till påverkan.

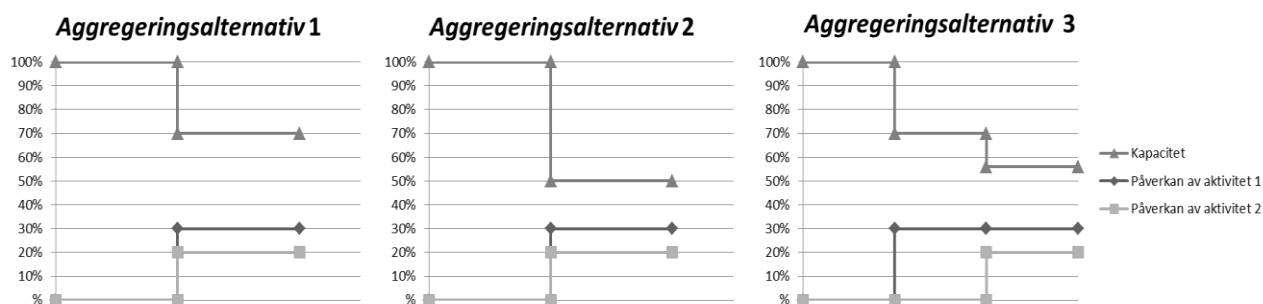
- Kopplat till föregående antagande kring linjäritet antas att påverkan och beroende är angivet i procent och inte i procentenheter. Därmed blir påverkan och beroende linjärt skalbart. Vid datainsamlingen tillfrågades aktörerna om hur deras aktiviteter blir påverkade vid ett flödesbortfall och svaren angavs i procentuell kapacitetsänkning. Aktörerna tog ställning till bortfall av enskilda flöden var för sig och det kan antas att aktören svarat utifrån full kapacitet. Däremot saknas uppgifter kring om svaren även är giltiga oberoende om aktuell kapacitet är nedsatt eller inte. Genom att anta att svaren är giltiga även för nedsatt kapacitet tolkas svaren som procent och inte procentenheter.

Tolkningen med procent kan beskrivas som att påverkan och beroende kan skalas med hjälp av aktuell kapacitet. Följande jämförelse av tolkningarnas effekt på en redan nedsatt aktivitet tydliggör detta och visar att tolkningsalternativen kan innebära betydande skillnader; en aktivitet har 40 % kapacitet och är 1-25 % (beräknas med medelvärdet 13 %) beroende av ett utslaget flöde. Om aktören angett svaret i procentenheter blir den kvarvarande kapaciteten 27 % (40 % - 13 %). Om aktören däremot angett svaret i procent skalas påverkan med hjälp av kapaciteten och den kvarvarande kapaciteten blir 35 % (40 % - 13 % * 40 %). Tolkningen i modellen är att aktören syftar på procent eftersom detta anses vara rimligare generellt. I vissa fall är tolkningen med procentenheter rimligare men det kan samtidigt innebära stora orimligheter i andra. Antagandet att påverkan och beroende är angivet i procent motiveras ytterligare i 5.2.2.

- Påverkan på en aktivitet från flera flöden samtidigt hanteras genom multiplikation och inte med hjälp av enskilt största påverkan eller addition av påverkan, se motivering i kapitel 5.2. Detta antagande är kopplat till det föregående men kräver ytterligare förklaring som presenteras här. Hanteringen av påverkan från flera aktiviteter samtidigt är problematisk bland annat på grund av begränsningar vid datainsamlingen. När data samlades in frågades inte aktörerna om hur bortfall av flera flöden påverkade dem utan enbart om påverkan från ett enskilt flöde i taget. Detta gör att hanteringen av påverkan från flera flöden samtidigt måste baseras på antaganden. Motsvarande gäller även för påverkan på flöden från flera aktiviteter. Tre möjliga aggregeringsalternativ presenteras i form av ett exempel i figur 7. I samtliga delfigurer är kapaciteten ursprungligen 100 % och aktivitet 1 och 2 påverkar flödet med 30 respektive 20 %. I *aggregeringsalternativ 1* sker påverkan baserat på största enskilda påverkan. *Aggregeringsalternativ 1* bör tolkas som att de två aktiviteterna påverkar precis samma del av ett flöde. Kan aktören bara sänka flödet med 30 % och detta är gjort av aktivitet 1 kan aktivitet 2 därmed inte påverka flödet alls. Generellt är detta aggregeringsalternativ inte lämpligt eftersom det inte går att säkerställa att bara en viss del av ett flöde kan påverkas, framförallt när antalet flöden som påverkar aktiviteten ökar.

Bortsett från största enskilda påverkan finns två möjligheter för hur aktörerna har angett sin påverkan, antingen att aktiviteten påverkar flödet med procentenheter eller procent. Om aktören har tolkat frågan som procentenheter kan påverkan aggregeras som för *aggregeringsalternativ 2* i figur 7, där påverkan från aktiviteterna adderas och summan subtraheras sedan från kapaciteten. Om aktören har tolkat frågan som procent kan påverkan aggregeras som för *aggregeringsalternativ 3* i figur 7, där först kvarvarande kapacitet efter aktivitet 1 beräknas och därefter multipliceras det återstående flödet med kvarvarande kapacitet efter påverkan av aktivitet 2.

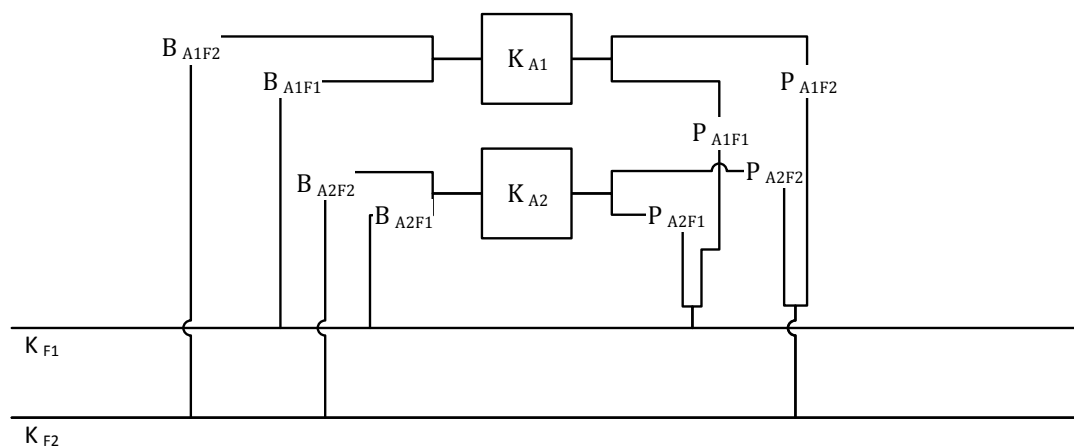
Resultatet, från de tre olika aggregeringsalternativen i figur 7, ger i första fallet en flödeskapacitet på 70 %, i andra 50 % och i det tredje 56 %. Alternativ 2 och 3 diskuteras mer ingående i nästföljande kapitel. Alternativ 1 utvärderas inte vidare då det inte finns något i tillgänglig indata som antyder att alternativet är lämpligt.



Figur 7. Aggregering av påverkan baserat på största enskilda, adderad eller multiplicerad påverkan. X-axeln visar enbart ordningen på aktiviteternas påverkan.

5.2 Metoder för att beräkna beroende och påverkan

De antaganden som presenterats i föregående kapitel har varierande betydelse för resultatet och är därmed i olika behov av att utvärderas ytterligare. Antagandet om hur påverkan från flera flöden samtidigt ska slås samman är en central del av modellen som används i examensarbetet och kräver ytterligare utvärdering. För att visa på komplexiteten och senare motivera valet av beräkningsmetod presenteras här en härledning av en beräkningsmetod baserad på addition och en baserad på multiplikation. Figur 8 visar en schematisk bild över ett förhållandevis enkelt system med två aktiviteter och två flöden. För att koppla exemplet till begreppet aktörer, kan aktiviteterna anses tillhöra var sin aktör. I tabell 5 presenteras de variabelnamn som används för att demonstrera beräkningsmetoderna. Beroendet och påverkan ska tolkas som hur starkt förhållandet mellan aktiviteten och flödet är, där 1 betyder fullständigt beroende eller fullständig påverkan. Kapacitetsvärdena visar funktionsnivån för en aktivitet respektive ett flöde, där 1 betyder fullständig funktion och 0 betyder fullständig utslagen.



Figur 8. Ett exempelsystem med två aktiviteter och två flöden.

Tabell 5. Variabelnamn samt beskrivning.

Variabelnamn	Beskrivning
$B_{A_1F_1}$	Beroende Aktivitet 1 av Flöde 1
$B_{A_1F_2}$	Beroende Aktivitet 1 av Flöde 2
$B_{A_2F_1}$	Beroende Aktivitet 2 av Flöde 1
$B_{A_2F_2}$	Beroende Aktivitet 2 av Flöde 2
$P_{A_1F_1}$	Påverkan Aktivitet 1 på Flöde 1
$P_{A_1F_2}$	Påverkan Aktivitet 1 på Flöde 2
$P_{A_2F_1}$	Påverkan Aktivitet 2 på Flöde 1
$P_{A_2F_2}$	Påverkan Aktivitet 2 på Flöde 2

$K_{A_1}^0$	<i>Kapacitet för Aktivitet 1 vid iteration 0</i>
$K_{A_2}^0$	<i>Kapacitet för Aktivitet 2 vid iteration 0</i>
$K_{F_1}^0$	<i>Kapacitet för Flöde 1 vid iteration 0</i>
$K_{F_2}^0$	<i>Kapacitet för Flöde 2 vid iteration 0</i>

Figur 8 kan användas för att exemplifiera spridningen av påverkan på en aktivitet. Spridningen av en störning börjar med att flöde 1 får en reduktion på 80 %, det vill säga att den får en kapacitet på 0,2. Detta påverkar aktivitet 1 och 2 i den utsträckning som aktiviteterna är beroende av flöde 1. Flödena i sin tur blir påverkade av aktiviteternas funktionsnivå. Det bildas alltså ett cirkelberoende, flödet blir indirekt påverkat av att den egna kapaciteten sjunker. I båda beräkningsmetoderna används en viss kapacitetstörning bara en gång. Detta genom att endast påverkan från den föregående iterationen används vid beräkningar i nästa iteration. Genom att endast kapacitetsförändringen från föregående beräkningsiteration används stabiliseras systemet vid oändligt många iterationer. Hade istället den totala kapacitetsnedsättningen använts vid varje iteration hade störningen snabbt ökat och samtliga aktiviteter och flöden hade slagits ut helt. Ekvation 5.1 visar hur kapacitetsförändringen beräknas för ett flöde och en aktivitet. Multiplikationen med ursprungskapaciteten i slutet av ekvation 5.1 är en följd av skalningsantagandet. Eftersom det antas att fullständig linjäritet råder påverkas aktiviteterna även av ett mindre flödesbortfall. För att beräkna hur mycket ett flöde påverkar en aktivitet multipliceras flödesbortfallet med beroendegraden.

$$\begin{aligned}\Delta K_{F_1}^1 &= (\Delta K_{A_1}^0 \cdot P_{A_1 F_1}) \cdot K_{F_1}^0 \\ \Delta K_{A_1}^1 &= (\Delta K_{F_1}^0 \cdot B_{A_1 F_1}) \cdot K_{A_1}^0\end{aligned}\quad 5.1$$

$\Delta K_{A_1}^1$ skall utläsas som kapacitetsförändringen för aktivitet 1 efter iteration 1.

5.2.1 Beräkningsmetod 1 – Additionsmetoden

Additionsmetoden bygger på aggregeringsalternativ 2 i figur 7. Den totala kapacitetsförändringen för aktivitet n kan beräknas genom att addera förändringen som flöde i och j ger upphov till.

Ursprungskapaciteten, $K_{A_n}^0$, är den samma för både flöde i och j , därmed kan den multipliceras med den totala förändringen, se ekvation 5.2.

$$\begin{aligned}\Delta K_{A_n}^1 &= (\Delta K_{F_i}^0 \cdot B_{A_n F_i} + \Delta K_{F_j}^0 \cdot B_{A_n F_j}) \cdot K_{A_n}^0 \\ K_{A_n}^1 &= \Delta K_{A_n}^0 - \Delta K_{A_n}^1\end{aligned}\quad 5.2$$

På motsvarande vis som aktiviteternas kapacitet beräknas kan flödets kapacitet beräknas enligt ekvation 5.3.

$$\begin{aligned}\Delta K_{F_i}^1 &= (\Delta K_{A_n}^1 \cdot P_{A_n F_i} + \Delta K_{A_m}^1 \cdot P_{A_m F_i}) \cdot K_{F_i}^0 \\ K_{F_i}^1 &= K_{F_i}^0 - \Delta K_{F_i}^1\end{aligned}\quad 5.3$$

Ekvation 5.3 är beroende av ekvation 5.2 då $\Delta K_{A_n}^1$ är resultatet av ekvation 5.2. För att beskriva kapaciteten med bara en ekvation kan ekvation 5.3 skrivas om enligt ekvation 5.4.

$$\begin{aligned}\Delta K_{F_i}^1 &= \left(\left[(\Delta K_{F_i}^0 \cdot B_{A_n F_i} + \Delta K_{F_j}^0 \cdot B_{A_n F_j}) \cdot K_{A_n}^0 \right] \cdot P_{A_n F_i} + \left[(\Delta K_{F_i}^0 \cdot B_{A_m F_i} + \Delta K_{F_j}^0 \cdot B_{A_m F_j}) \cdot K_{A_m}^0 \right] \cdot P_{A_m F_i} \right) \cdot K_{F_i}^0 \\ K_{F_i}^1 &= K_{F_i}^0 - \Delta K_{F_i}^1\end{aligned}\quad 5.4$$

Ekvation 5.4 är det sista steget för iteration 1 och eftersom såväl aktiviteter som flöden kan ha blivit påverkade under iterationen behöver deras förändrade kapaciteter beräknas. Med förändrade kapaciteter behöver ekvation 5.2 räknas om, vilket visas i ekvation 5.5.

$$\begin{aligned} \Delta K_{A_n}^2 &= \left(\Delta K_{F_i}^1 \cdot B_{A_n F_i} + \Delta K_{F_j}^1 \cdot B_{A_n F_j} \right) \cdot K_{A_n}^1 & 5.5 \\ K_{A_n}^2 &= \Delta K_{A_n}^1 - \Delta K_{A_n}^2 \end{aligned}$$

Kapacitetsförändringen för aktiviteterna efter iteration 2 påverkar flödena i ekvation 5.6.

$$\begin{aligned} \Delta K_{F_i}^2 &= \left(\left[\left(\Delta K_{F_i}^1 \cdot B_{A_n F_i} + \Delta K_{F_j}^1 \cdot B_{A_n F_j} \right) \cdot K_{A_n}^1 \right] \cdot P_{A_n F_i} + \left[\left(\Delta K_{F_i}^1 \cdot B_{A_m F_i} + \Delta K_{F_j}^1 \cdot B_{A_m F_j} \right) \cdot K_{A_m}^1 \right] \cdot P_{A_m F_i} \right) \cdot K_{F_i}^1 & 5.6 \\ K_{F_i}^2 &= K_{F_i}^1 - \Delta K_{F_i}^2 \end{aligned}$$

Kapacitetsförändringen för flödena efter iteration 2 påverkar aktiviteterna i iteration 3. Detta påverkar flödena och iterationerna kommer fortsätta till samtliga förändringar är mindre än 1 %, enligt antagandet om iterationer i kapitel 5.1.2.

Följande exempel kan användas för att belysa hur additionsmetoden fungerar; en aktör har vid datainsamlingen angett sitt beroende av fyra flöden till 1-25 % och om det övre värdet i intervallet, 25 %, används kommer ett fullständigt bortfall av de fyra flödena slå ut aktören fullständigt. Att fyra fall av något som vid datainsamlingen angetts som marginell påverkan kan resultera i fullständig utslagning kan uppfattas som orimligt och kan ge kraftiga överskattningar av påverkan när det inte är säkerställt att påverkan kan adderas.

5.2.2 Beräkningsmetod 2 – Multiplikationsmetoden

Multiplikationsmetoden bygger på aggregeringsalternativ 3 i figur 7. Den totala kapacitetsförändringen för aktivitet n kan beräknas genom att multiplicera störningen som flöde i och j ger upphov till. Om multiplikationen genomförs direkt på störningen kommer den totala påverkan bli noll då en aktivitet inte är nedsatt eller om aktiviteten är oberoende av flödet. För att inte kapaciteten ska bli noll och störningen ska bli större med fler påverkande, måste ekvation 5.2 modifieras utöver att multiplikation ersätter additionen. Förändringen är att istället för att multiplicera termerna direkt så subtraheras störningen, från respektive flöde, från ett innan multiplikationen, se ekvation 5.7.

$$\begin{aligned} \Delta K_{A_n}^1 &= K_{A_n}^0 - \left((1 - \Delta K_{F_i}^0 \cdot B_{A_n F_i}) \cdot (1 - \Delta K_{F_j}^0 \cdot B_{A_n F_j}) \right) \cdot K_{A_n}^0 & 5.7 \\ K_{A_n}^1 &= \Delta K_{A_n}^0 - \Delta K_{A_n}^1 \end{aligned}$$

På motsvarande vis som aktiviteternas kapacitet beräknas kan flödets kapacitet beräknas enligt ekvation 5.8.

$$\begin{aligned} \Delta K_{F_i}^1 &= K_{F_i}^0 - \left((1 - \Delta K_{A_n}^1 \cdot P_{A_n F_i}) \cdot (1 - \Delta K_{A_m}^1 \cdot P_{A_m F_i}) \right) \cdot K_{F_i}^0 & 5.8 \\ K_{F_i}^1 &= K_{F_i}^0 - \Delta K_{F_i}^1 \end{aligned}$$

På motsvarande sätt som ekvation 5.3 är beroende av ekvation 5.4 är ekvation 5.8 beroende av ekvation 5.7. För att beskriva kapaciteten med en ekvation kan ekvation 5.8 skrivas om till ekvation 5.9.

$$\begin{aligned} \Delta K_{F_i}^1 &= K_{F_i}^0 - \left(\left(1 - \left[K_{A_n}^0 - \left((1 - \Delta K_{F_i}^0 \cdot B_{A_n F_i}) \cdot (1 - \Delta K_{F_j}^0 \cdot B_{A_n F_j}) \right) \cdot K_{A_n}^0 \right] \cdot P_{A_n F_i} \right) \cdot \right. \\ &\quad \left. \left(1 - \left[K_{A_m}^0 - \left((1 - \Delta K_{F_i}^0 \cdot B_{A_m F_i}) \cdot (1 - \Delta K_{F_j}^0 \cdot B_{A_m F_j}) \right) \cdot K_{A_m}^0 \right] \cdot P_{A_m F_i} \right) \right) \cdot K_{F_i}^0 \\ K_{F_i}^1 &= K_{F_i}^0 - \Delta K_{F_i}^1 \end{aligned} \quad 5.9$$

Ekvation 5.9 är sista steget för iteration 1 och eftersom både aktiviteter och flöden kan ha blivit påverkade under iterationen behöver kapaciteterna beräknas på nytt. Med förändrade flödeskapaciteter behöver ekvationer 5.6 räknas om, vilket visas i ekvation 5.10.

$$\begin{aligned} \Delta K_{A_n}^2 &= K_{A_n}^1 - \left((1 - \Delta K_{F_i}^1 \cdot B_{A_n F_i}) \cdot (1 - \Delta K_{F_j}^1 \cdot B_{A_n F_j}) \right) \cdot K_{A_n}^1 \\ K_{A_n}^2 &= \Delta K_{A_n}^1 - \Delta K_{A_n}^2 \end{aligned} \quad 5.10$$

Kapacitetsförändringen för aktiviteterna efter iteration 2 påverkar flödena i ekvation 5.11.

$$\begin{aligned} \Delta K_{F_i}^2 &= K_{F_i}^1 - \left(\left(1 - \left[K_{A_n}^1 - \left((1 - \Delta K_{F_i}^1 \cdot B_{A_n F_i}) \cdot (1 - \Delta K_{F_j}^1 \cdot B_{A_n F_j}) \right) \cdot K_{A_n}^1 \right] \cdot P_{A_n F_i} \right) \cdot \right. \\ &\quad \left. \left(1 - \left[K_{A_m}^1 - \left((1 - \Delta K_{F_i}^1 \cdot B_{A_m F_i}) \cdot (1 - \Delta K_{F_j}^1 \cdot B_{A_m F_j}) \right) \cdot K_{A_m}^1 \right] \cdot P_{A_m F_i} \right) \right) \cdot K_{F_i}^1 \\ K_{F_i}^2 &= K_{F_i}^1 - \Delta K_{F_i}^2 \end{aligned} \quad 5.11$$

Kapacitetsförändringen för flödena efter iteration 2 påverkar aktiviteterna i iteration 3. Detta påverkar flödena och iterationerna kommer fortsätta till samtliga förändringar är mindre än 1 %, enligt antagandet om iterationer i kapitel 5.1.2.

Om samma exempel som användes för additionsmetoden, med fyra flöden som påverkade en aktivitet med 25 %, används för multiplikationsmetoden kan den kvarvarande kapaciteten beräknas till:

$$\text{Kapacitet efter påverkan} = (1 - 0,25)^4 = 0,75^4 = 0,56$$

Med samma resonemang som när exemplet användes för additionsmetoden kan det anses vara rimligt att en aktivitet har 56 % kapacitet när den har påverkats marginellt fyra gånger. Samtidigt innebär tolkningen att svaret blir orimligt om aktören syftar på påverkan i form av procentenheter.

5.2.3 Beräkningsexempel

För att tydliggöra skillnaderna mellan beräkningsmetoderna beräknas exemplet från figur 8 med de slumpmässiga värden som presenteras i tabell 6.

Tabell 6. Beskrivning och data för exemplet.

Variabelnamn	Beskrivning	Värde
$B_{A_1 F_1}$	Beroende Aktivitet 1 av Flöde 1	0,60
$B_{A_1 F_2}$	Beroende Aktivitet 1 av Flöde 2	0,80
$B_{A_2 F_1}$	Beroende Aktivitet 2 av Flöde 1	0,70
$B_{A_2 F_2}$	Beroende Aktivitet 2 av Flöde 2	0,38
$P_{A_1 F_1}$	Påverkan Aktivitet 1 på Flöde 1	0,10
$P_{A_1 F_2}$	Påverkan Aktivitet 1 på Flöde 2	0,30
$P_{A_2 F_1}$	Påverkan Aktivitet 2 på Flöde 1	0,40
$P_{A_2 F_2}$	Påverkan Aktivitet 2 på Flöde 2	0,80
$K_{A_1}^0$	Kapacitet för Aktivitet 1 vid iteration 0	1,0
$K_{A_2}^0$	Kapacitet för Aktivitet 2 vid iteration 0	1,0

$K_{F_1}^0$	Kapacitet för Flöde 1 vid iteration 0	0,20
$K_{F_2}^0$	Kapacitet för Flöde 2 vid iteration 0	1,0

Beräknas aktivitetens kapacitet med additionsmetoden, ekvation 5.2, blir resultatet följande:

$$\begin{aligned}\Delta K_{A_1}^1 &= ((1 - 0,2) \cdot 0,6 + (1 - 1) \cdot 0,8) \cdot 1 = 0,48 & 5.12 \\ K_{A_1}^1 &= \Delta K_{A_1}^0 - \Delta K_{A_1}^1 = 0,52\end{aligned}$$

$$\begin{aligned}\Delta K_{A_2}^1 &= ((1 - 0,2) \cdot 0,7 + (1 - 1) \cdot 0,38) \cdot 1 = 0,56 & 5.13 \\ K_{A_2}^1 &= \Delta K_{A_2}^0 - \Delta K_{A_2}^1 = 0,44\end{aligned}$$

Beräknas aktivitetens kapacitet med multiplikationsmetoden, ekvation 5.7 blir resultatet följande:

$$\Delta K_{A_1}^1 = 1 - (1 - (1 - 0,2) \cdot 0,6 \cdot (1 - (1 - 1) \cdot 0,8)) \cdot 1 = 0,48 \quad 5.14$$

$$K_{A_1}^1 = \Delta K_{A_1}^0 - \Delta K_{A_1}^1 = 0,52$$

$$\Delta K_{A_2}^1 = 1 - (1 - (1 - 0,2) \cdot 0,7 \cdot (1 - (1 - 1) \cdot 0,38)) \cdot 1 = 0,56 \quad 5.15$$

$$K_{A_2}^1 = \Delta K_{A_2}^0 - \Delta K_{A_2}^1 = 0,44$$

Resultatet av ekvation 5.10 och 5.11 är identiskt med ekvation 5.14 och 5.15 eftersom $\Delta K_{F_2}^0 (1 - 1) = 0$, vilket betyder att flöde 2 inte påverkar aktiviteterna under iteration 1 och ingen sammanslagning har genomförts. Beräkningsmetoderna skiljer sig enbart vid sammanslagningar av påverkan och ger därför samma resultat. Beräkningsmetoderna kommer däremot att skilja sig åt för resterande ekvationer.

Flödets kapacitet kan beräknas med additionsmetoden enligt ekvation 5.4:

$$\begin{aligned}\Delta K_{F_1}^1 &= \left(((1 - 0,2) \cdot 0,6 + (1 - 1) \cdot 0,8) \cdot 1 \cdot 0,1 + ((1 - 0,2) \cdot 0,7 + (1 - 1) \cdot 0,38) \cdot \right. & 5.16 \\ &\left. 1 \cdot 0,4 \right) \cdot 0,2 = 0,0544\end{aligned}$$

$$K_{F_1}^1 = 0,1456$$

$$\begin{aligned}\Delta K_{F_2}^1 &= \left(((1 - 0,2) \cdot 0,6 + (1 - 1) \cdot 0,8) \cdot 1 \cdot 0,3 + ((1 - 0,2) \cdot 0,7 + (1 - 1) \cdot 0,38) \cdot \right. & 5.17 \\ &\left. 1 \cdot 0,2 \right) \cdot 1 = 0,592\end{aligned}$$

$$\Delta K_{F_2}^1 = 0,408$$

Enligt multiplikationsmetoden beräknas flödenas kapacitet med ekvation 5.9 till:

$$\Delta K_{F_1}^1 = 0,2 - \left((1 - [1 - ((1 - (1 - 0,2) \cdot 0,6) \cdot (1 - (1 - 1) \cdot 0,8))] \cdot 1) \cdot 0,1) \cdot \right. & 5.18$$

$$\left. (1 - [1 - ((1 - (1 - 0,2) \cdot 0,7) \cdot (1 - (1 - 1) \cdot 0,38))] \cdot 1) \cdot 0,4) \right) \cdot 0,2 = 0,0523$$

$$K_{F_1}^1 = 0,1478$$

$$\Delta K_{F_2}^1 = 1 - \left((1 - [1 - ((1 - (1 - 0,2) \cdot 0,6) \cdot (1 - (1 - 1) \cdot 0,8))] \cdot 1) \cdot 0,3) \cdot \right. & 5.19$$

$$\left. (1 - [1 - ((1 - (1 - 0,2) \cdot 0,7) \cdot (1 - (1 - 1) \cdot 0,38))] \cdot 1) \cdot 0,2) \right) \cdot 1 = 0,5275$$

$$\Delta K_{F_2}^1 = 0,4725$$

Aktivitetens kapacitet efter iteration 2 beräknas enligt följande med additionsmetodens ekvation 5.5:

$$\Delta K_{A_1}^2 = (0,0544 \cdot 0,6 + 0,592 \cdot 0,8) \cdot 0,52 = 0,2632 \quad 5.20$$

$$K_{A_1}^2 = 0,2568$$

$$\Delta K_{A_2}^2 = (0,0544 \cdot 0,7 + 0,592 \cdot 0,38) \cdot 0,44 = 0,1160 \quad 5.21$$

$$K_{A_2}^2 = 0,3242$$

Aktivitetens kapacitet efter iteration 2 beräknas enligt multiplikationsmetoden med ekvation 5.10 som följande:

$$\Delta K_{A_1}^2 = 0,52 - ((1 - 0,0522 \cdot 0,6) \cdot (1 - 0,5275 \cdot 0,8)) \cdot 0,52 = 0,2289 \quad 5.22$$

$$K_{A_1}^2 = 0,2911$$

$$\Delta K_{A_1}^2 = 0,44 - ((1 - 0,0522 \cdot 0,7) \cdot (1 - 0,5275 \cdot 0,38)) \cdot 0,44 = 0,1011 \quad 5.23$$

$$K_{A_2}^2 = 0,3389$$

Flödets kapacitet efter iteration 2 beräknas enligt följande med additionsmetodens ekvation 5.6:

$$\Delta K_{F_1}^2 = \left(((0,0544) \cdot 0,6 + (0,592) \cdot 0,8) \cdot 0,52 \cdot 0,1 + ((0,0544) \cdot 0,7 + (0,592) \cdot 0,38) \cdot 0,44 \cdot 0,4 \right) \cdot 0,1456 = 0,0106 \quad 5.24$$

$$K_{F_1}^2 = 0,1350$$

$$\Delta K_{F_2}^2 = \left(((0,0544) \cdot 0,6 + (0,592) \cdot 0,8) \cdot 0,52 \cdot 0,3 + ((0,0544) \cdot 0,7 + (0,592) \cdot 0,38) \cdot 0,44 \cdot 0,2 \right) \cdot 0,4080 = 0,070 \quad 5.25$$

$$K_{F_2}^2 = 0,3380$$

Flödets kapacitet efter iteration 2 beräknas enligt multiplikationsmetoden med ekvation 5.12 som följande:

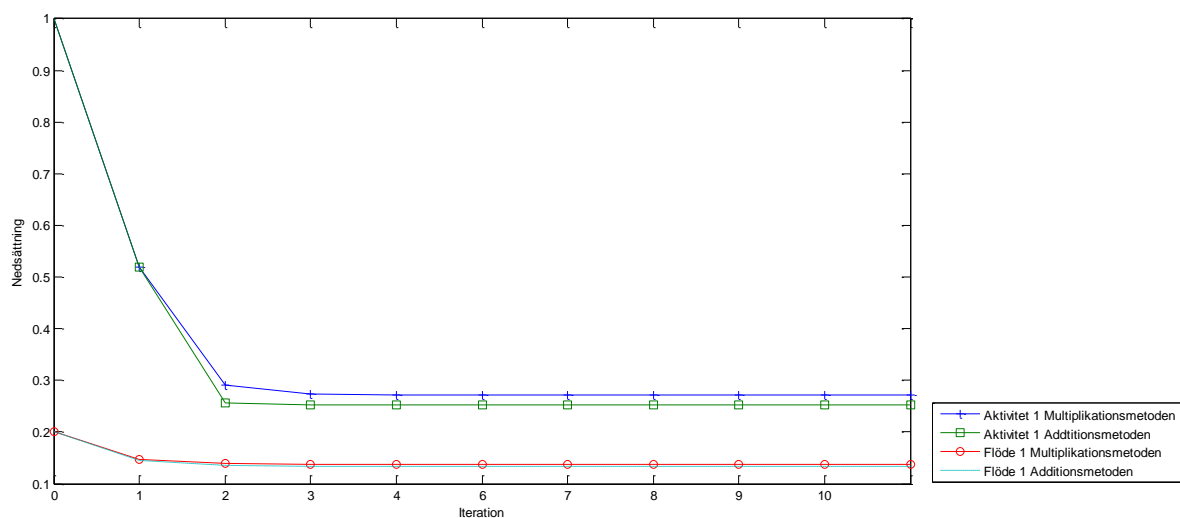
$$\Delta K_{F_1}^2 = 0,1478 - \left((1 - [0,2911 - ((1 - (0,0523) \cdot 0,6) \cdot (1 - (0,5275) \cdot 0,8)) \cdot 0,2911] \cdot 0,1) \cdot (1 - [0,3389 - ((1 - (0,0523) \cdot 0,7) \cdot (1 - (0,5275) \cdot 0,38)) \cdot 0,3389] \cdot 0,4) \right) \cdot 0,1478 = 0,0093 \quad 5.26$$

$$K_{F_1}^2 = 0,1385$$

$$\Delta K_{F_2}^2 = 0,4725 - \left((1 - [0,2911 - ((1 - (0,0523) \cdot 0,6) \cdot (1 - (0,5275) \cdot 0,8)) \cdot 0,2911] \cdot 0,3) \cdot (1 - [0,3389 - ((1 - (0,0523) \cdot 0,7) \cdot (1 - (0,5275) \cdot 0,38)) \cdot 0,3389] \cdot 0,8) \right) \cdot 0,4725 = 0,0681 \quad 5.27$$

$$K_{F_2}^2 = 0,4044$$

Iteration 3 följer samma mönster som iteration 2 men med resultaten från iteration 2 som ingångsvärde. Figur 9 visar kapacitetsstörningar beräknat med de både multiplikationsmetoden och additionsmetoden. I figuren syns det att det inte sker någon markant sänkning efter den tredje iterationen. Med fortsatta iterationsberäkningar uppfylldes antagandet om att avbryta iterationerna då påverkan är mindre än 1 % efter den fjärde iterationen. I figuren syns också att det inte är någon större skillnad mellan beräkningsmetoderna när de används för beräkningsexemplet.

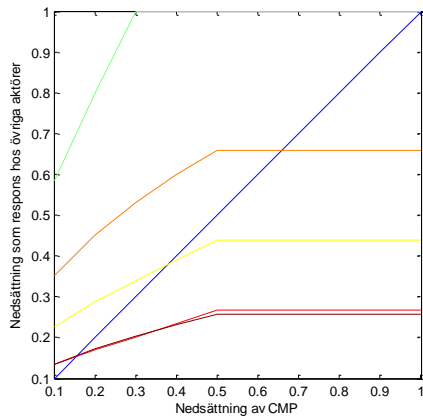


Figur 9. Kapaciteten för aktivitet 1 och flöde 1 beräknat, i tio beräkningsiterationer, dels med multiplikationsmetoden dels med additionsmetoden.

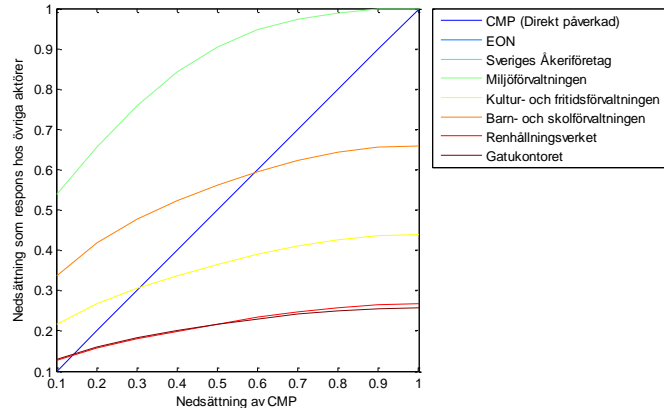
5.2.4 Summering och val av beräkningsmetod

De två beräkningsmetoderna ger liknande resultat när exemplet ovan beräknas och båda metoderna anses uppfylla kraven baserat på målen för modellen och tillgänglig indata. Efter tio iterationer blir skillnaden störst för flöde 2 där det skiljer drygt 6 procentenheter mellan additions och multiplikationsmetoden, se ekvation 5.25 och 5.27. Exemplet visar att multiplikationsmetoden ger högre kapaciteter för alla flöden och alla aktiviteter, vilket även indikeras i figur 7 och visas i figur 9.

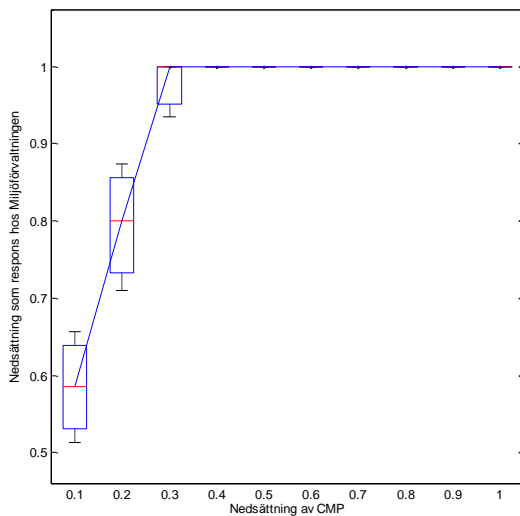
Det går att jämföra de två beräkningsmetoderna genom att visa hur responsen hos en eller alla aktörer ser ut när en förutbestämd aktör slås ut stegvis (J., Johansson, personlig kommunikation, 20 oktober, 2013). Exempelsystemet som används i kapitel 5.2 är för litet för att responsgraferna ska visa på skillnader mellan metoderna och därför redovisas istället grafer för annan indata, som presenteras i kapitel 6.2.7. Figur 10 och 11 visar hur en stegvis störning av aktören Copenhagen Malmö Port (CMP) påverkar störningen hos resterande aktörer. Generellt syns det att multiplikationsmetoden ger lägre respons i form av störningar än additionsmetoden även om båda metoderna leder till samma resultat vid fullständig utslagning av CMP. I figur 12 och 13 visas de beräknande störningarna baserat på det högsta och lägsta värde för beroende och påverkan som miljöförvaltningen och CMP har angett i samband med datainsamlingen. I dessa figurer är det tydligt att additionsmetoden ger kraftigare respons vid små störningar och att spridningen för responsen är större än för multiplikationsmetoden. Därmed kan det anses att additionsmetoden är mer konservativ eftersom kraftigare störningar uppnås tidigt och den visar på känsligheten till indata genom att ha större spridning i resultatet. Samtidigt är skillnaderna mellan metoderna inte stora vid fullständiga utslagningar och multiplikationsmetoden möjliggör analys av fler störningsscenarier eftersom responsen inte direkt går mot en stabil nivå.



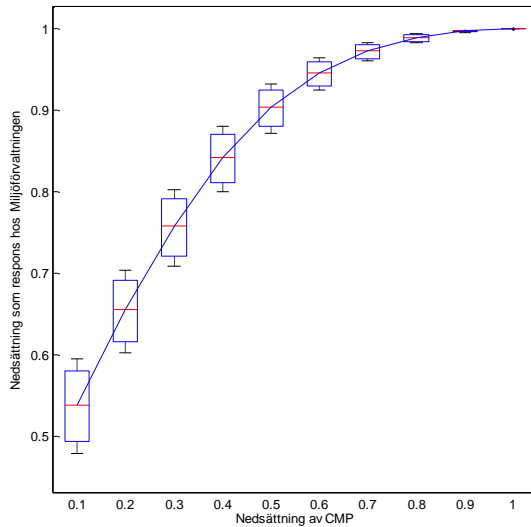
Figur 10. Exempel på respons för samtliga aktörer med additionsmetoden. Störning av CMP.



Figur 11. Exempel på respons för samtliga aktörer med multiplikationsmetoden. Störning av CMP.



Figur 12. Exempel på respons hos en aktör med additionsmetoden. Störning av CMP.



Figur 13. Exempel på respons hos en aktör med multiplikationsmetoden. Störning av CMP. Observera konvergensen i spridning inte gäller för samtliga fall.

En aspekt av beräkningsmetoderna som inte visas i exemplet i kapitel 5.2.3 är när två viktiga aktörer, för ett flöde, blir utslagna och den totala störningen för flödet därmed är större än 100 %. Detta kan till exempel hända om båda aktörerna angett att de har 100 % påverkan på ett flöde och de blir utslagna samtidigt. I multiplikationsmetoden kommer flödets kapacitet gå mot noll men i additionsmetoden finns det en möjlighet att kapaciteten går under noll eftersom den totala påverkan från varje iteration dras bort från kapaciteten. Kapaciteten för ett flöde eller en aktör kan aldrig vara under 0 % därför måste kapaciteten hindras från att gå under noll i additionsmetoden, vilket kan anses som en begränsning med metoden. Om additionsmetoden används behöver datormodellen ett villkor som förhindrar kapaciteten från att gå under noll. Samtidigt skulle detta villkor vara realistiskt eftersom det i verkligheten bör finnas flera enskilda flöden som kan sänka en aktivitet fullständigt.

Både att addera och att multiplicera påverkan har brister vid vissa beräkningssteg, vilket medför att resultaten kommer innehålla skillnader mot ett verkligt system. Oavsett vilken av metoderna som väljs kommer det att vara felaktig för viss data. Multiplikationsmetoden stämmer väl överens med antagandet som gjorts kring linjäritet mellan kapacitet och påverkan och kan därför anses vara lämpligare. Även baserat på hur datainsamling och svarsalternativen är formulerat och på antagandet om skalning bör multiplikationsmetoden vara den minst felaktiga och kommer därmed att användas i datorverktyget. Vid framtida utveckling av beräkningsmetoden är det lämpligt att genomföra en större studie med fokus på vilken beräkningsmetod som stämmer bäst med verkligheten.

6 Visualisering av resultatet

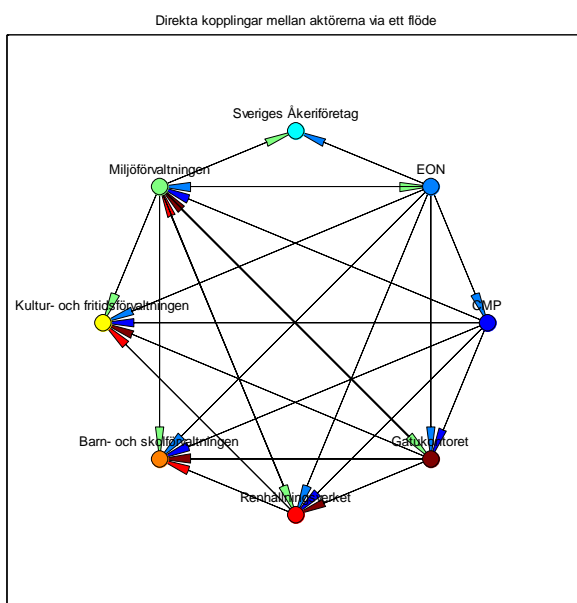
Datorverktyget utvecklades utifrån de mål som satts upp för examensarbetet och den kravspecifikation som togs fram tillsammans med framtida användare. De funktioner som innefattas av datorverktyget och dess visualiseringar presenteras i detta kapitel. För att presentera de utvecklade funktionerna används data från både Malmö och Lunds kommun i figurerna vilket innebär att den data som visas bara ska ses som exempel. En modellering av ett verkligt system presenteras istället i kapitel 7 där en fallsudie genomförs på Lunds kommun. Tillsammans med varje funktion presenteras vad som visas i respektive figur och till vilken grad funktionen överensstämmer med kravspecifikationen som presenterades i Appendix A. Funktionerna kan delas upp i sådana som visualiserar indata och sådana som visualiserar analysresultatet efter störningar och ger aktörers och flödens kapaciteter. Analysresultaten beräknas med hjälp av beräkningsmetoden som presenteras i kapitel 5.2.2. De flesta funktioner som presenteras är framtagna inom ramarna för examensarbetet men vissa av funktionerna som presenterar indata direkt är tidigare framtagna av examensarbetets ena handledare Jonas Johansson, LTH. Dessa presenteras bara kort för att ge fullständig bild av det framtagna datorverktygets möjligheter. När datorverktyget används går det att välja mellan vilka figurer som ska genereras genom att välja ett antal olika funktioner. För varje figur som presenteras i detta kapitel finns det alltså en specifik funktion.

6.1 Visualiseringar av indata

Visualiseringarna i detta kapitel är skapade utan beräkningar av störningsspridning. Figurerna visar alltså olika sätt som indata kan visualiseras på utan att beräkningsmetoden har använts. Ingen av visualiseringarna omfattas av kravspecifikationen utan de presenteras enbart för att visa datorverktygets samtliga funktioner.

6.1.1 Direkta kopplingar mellan aktörer via flöde

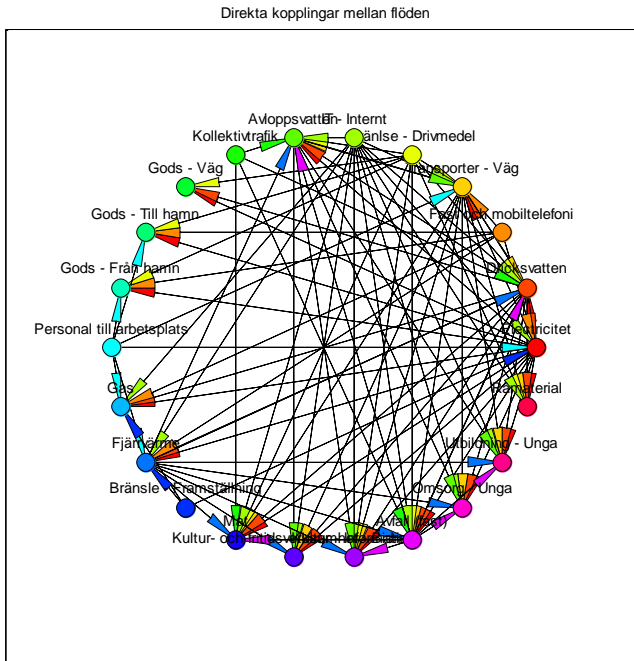
Funktionen skapar en figur som visar vilka direkta kopplingar som finns mellan aktörerna i någon av tidsperioderna, se figur 14. Funktionen skapades av handledaren, Jonas Johansson, LTH. Figuren har modifierats för att visa påverkan med hjälp pilar istället för linjer men är i övrigt i originalutförande.



Figur 14. Visualisering av aktörers direkta kopplingar via flöden. Pilarna går från påverkande till beroende aktör. Färgen på pilarna hämtas från de påverkande aktörerna.

6.1.2 Direkta kopplingar mellan flöden

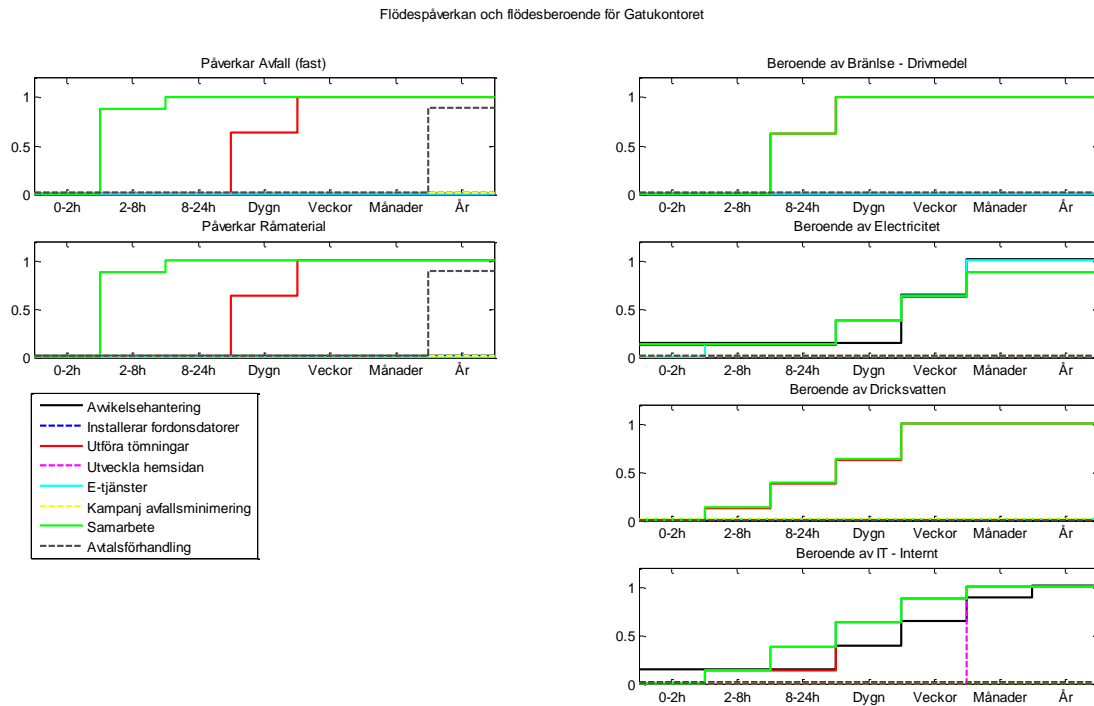
Funktionen skapar en figur som visar direkta kopplingar mellan flödena i någon av tidsperioderna, se figur 15. Att presentera direkta kopplingar i en cirkel är baserat på föregående figur som skapades av handledaren, Jonas Johansson, LTH och figuren nedan är bara modifierad för att hantera flöden.



Figur 15. Visualisering av direkta kopplingar mellan flöden. Pilarna går från påverkande till beroende flöde. Färgen på pilarna hämtas från de påverkande flödena.

6.1.3 Direkt flödespåverkan och flödesberoende för en aktör

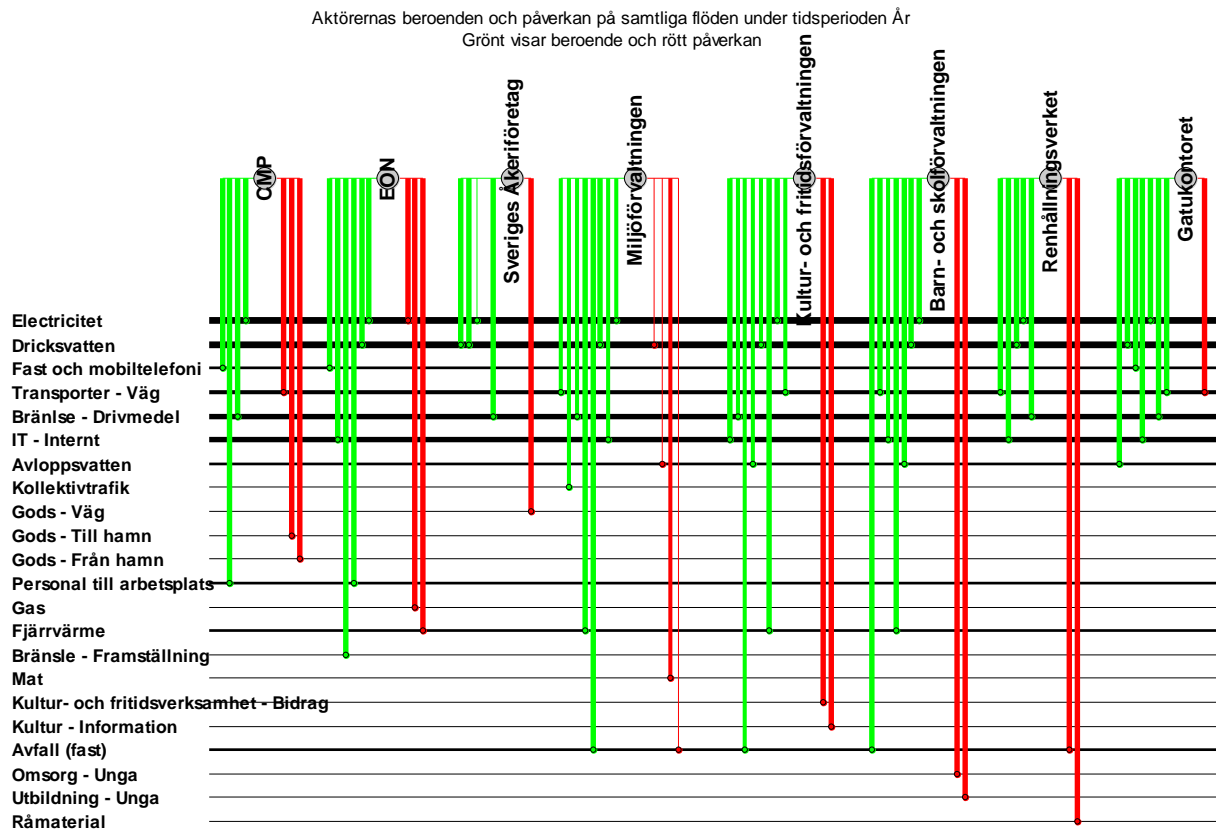
Figur 16 visar en aktörs påverkan på och beroende av flöden under varje tidsperiod. Figuren skapades av handledaren, Jonas Johansson, LTH.



Figur 16. Direkt flödespåverkan och beroende för en aktör under alla tidsperioder. 1 innebär fullständigt beroende och 0 innebär obefintligt beroende. Färgen på linjerna i delfigurerna motsvarar de aktiviteterna som presteras i teckenförklaringen.

6.1.4 Samtliga aktörers beroenden och påverkan under en tidsperiod

Funktionen visar vilka flöden varje aktör är beroende av och har påverkan på under en vald tidsperiod. Ett exempel visas i figur 17. Linjerna i figuren visar kopplingarna mellan aktörer och flöden samt deras styrka. Beroenden av flöden visas med gröna linjer. Tjockleken på linjerna bestäms av den av aktörens aktiviteter som är maximalt beroende av flödet. Påverkan på flöden visas med röda linjer och tjockleken bestäms med hjälp av den av aktörens aktiviteter som kan påverkas maximalt. Tjockleken på den svarta linje som går från respektive flöde bestäms av hur många aktörer som är beroende av flödet och styrkan på dessa beroenden. Figuren skapades av handledaren, Jonas Johansson, LTH.

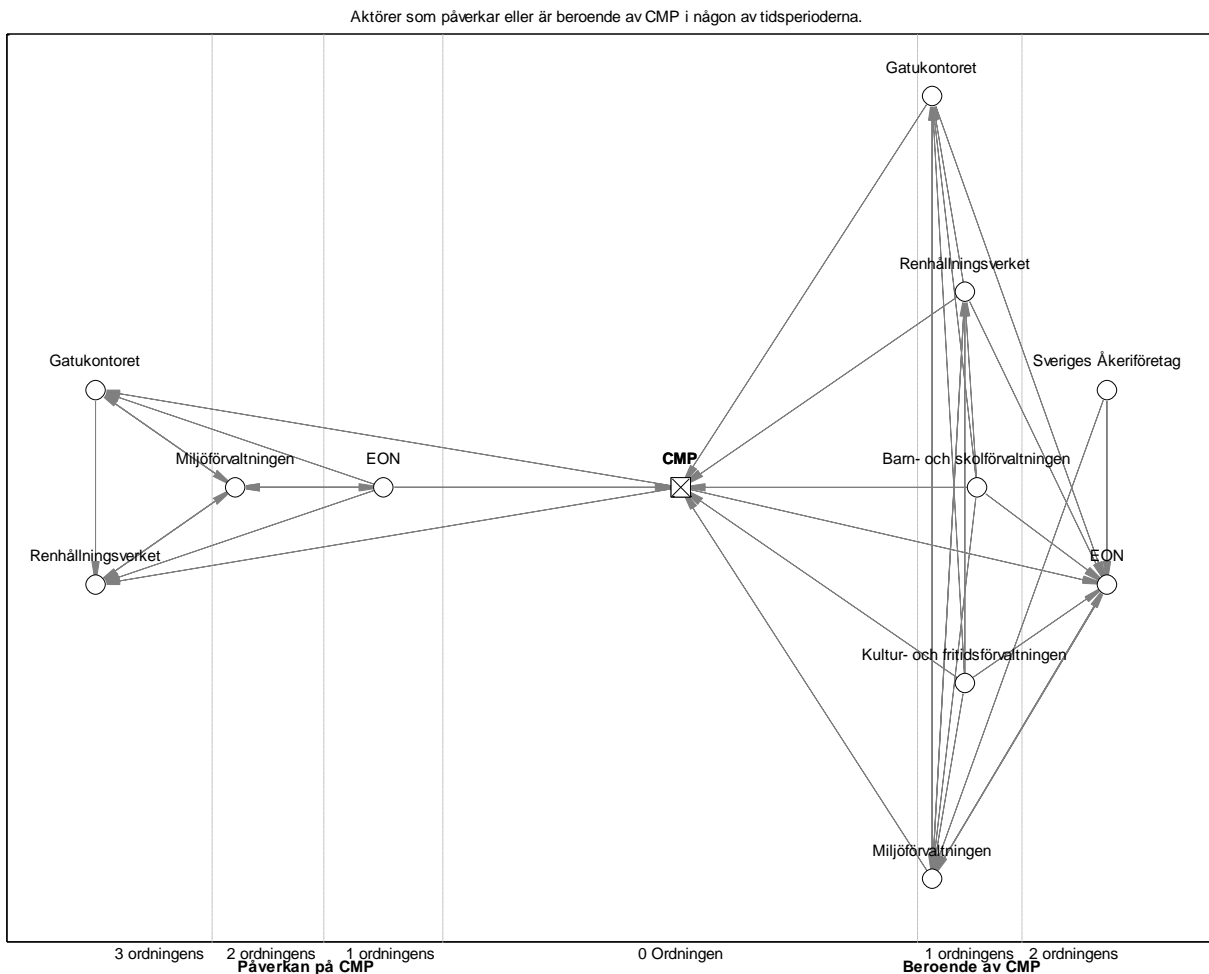


Figur 17. Aktörernas beroenden och påverkan på samtliga flöden under en viss tidsperiod. Grönt visar beroende av flödet och rött påverkan på flödet.

6.1.5 Spridning- och beroendekedjor för en aktör

Funktionen skapar en figur som visar vilka aktörer som blir direkt och indirekt påverkade vid utslagning av en vald aktör. Den visar också vilka aktörer den valda aktören är direkt och indirekt beroende av.

Figuren är baserad på beroende- och spridningskedjor som presenteras av MSB (2007) som en fortsättning på beroendeanalys baserad på beroendehjulet, se kapitel 2.6.5. I beroende- och spridningskedjor presenteras hur påverkan sprids i flera led. Liknande resonemang förs av Rinaldi et. al. (2001) men leden benämns som första, andra och n :t ordningens påverkan. Figur 18 visar ett exempel på dessa beroende- och spridningskedjor med benämningar från Rinaldi et. al. (2001) där aktören Copenhagen Malmö Port (CMP) är satt i mitten. Kopplingen mellan aktörerna är baserad på om den valda aktören påverkar ett flöde som de påverkade aktörerna är beroende av i något tidsintervall. Genom att spridningskedjor skapas för varje påverkad aktör kommer även aktörer med annat än första ordningens beroende av den utslagna aktören synas i figuren. Motsvarande gäller vänster del av figuren där det i figur 18 visas aktörer som har första, andra och tredje ordningens påverkan på CMP. Till vänster i figur 18 finns även exempel på tre feedback-loopar, en direkt och två indirekta. Den direkta feedback-loopen är mellan miljöförvaltningen och E.ON och illustreras i figuren med en dubbelpil. De indirekta feedback-looparna är att E.ON påverkar både gatukontoret och renhållningsverket som båda påverkar miljöförvaltningen som i sin tur påverkar E.ON.



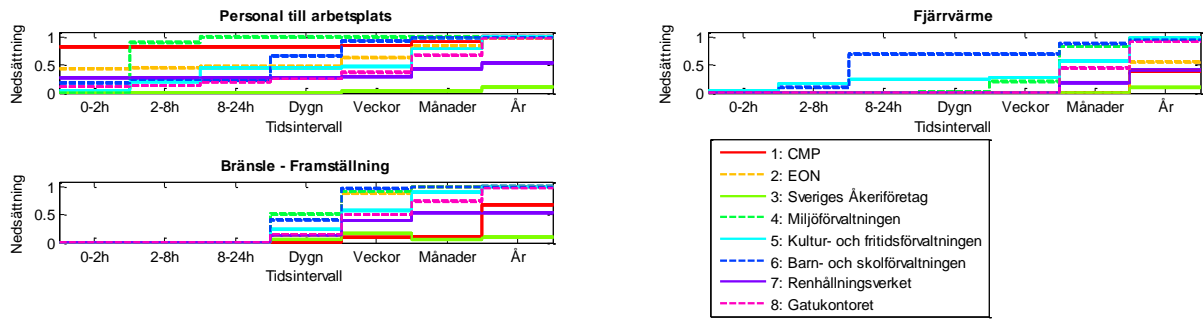
Figur 18. Spridning- och beroendekedjor för aktören Copenhagen Malmö Port. Pilarna går från påverkande till beroende aktör.

6.2 Visualisering av analysresultat

Samtliga funktioner i detta kapitel baseras på de antagande som presenterades i kapitel 5.1. Ekvationerna som ligger till grund för aktiviteternas påverkan och beroende av flöden presenterades i kapitel 5.2.2. Övergången från aktivitetens kapacitet till aktörers kapacitet sker med hjälp av viktning mellan aktiviteternas måluppfyllnad som är beskrivet under kapitel 5.2.2.

6.2.1 Aktörers fullständiga beroende av flöden

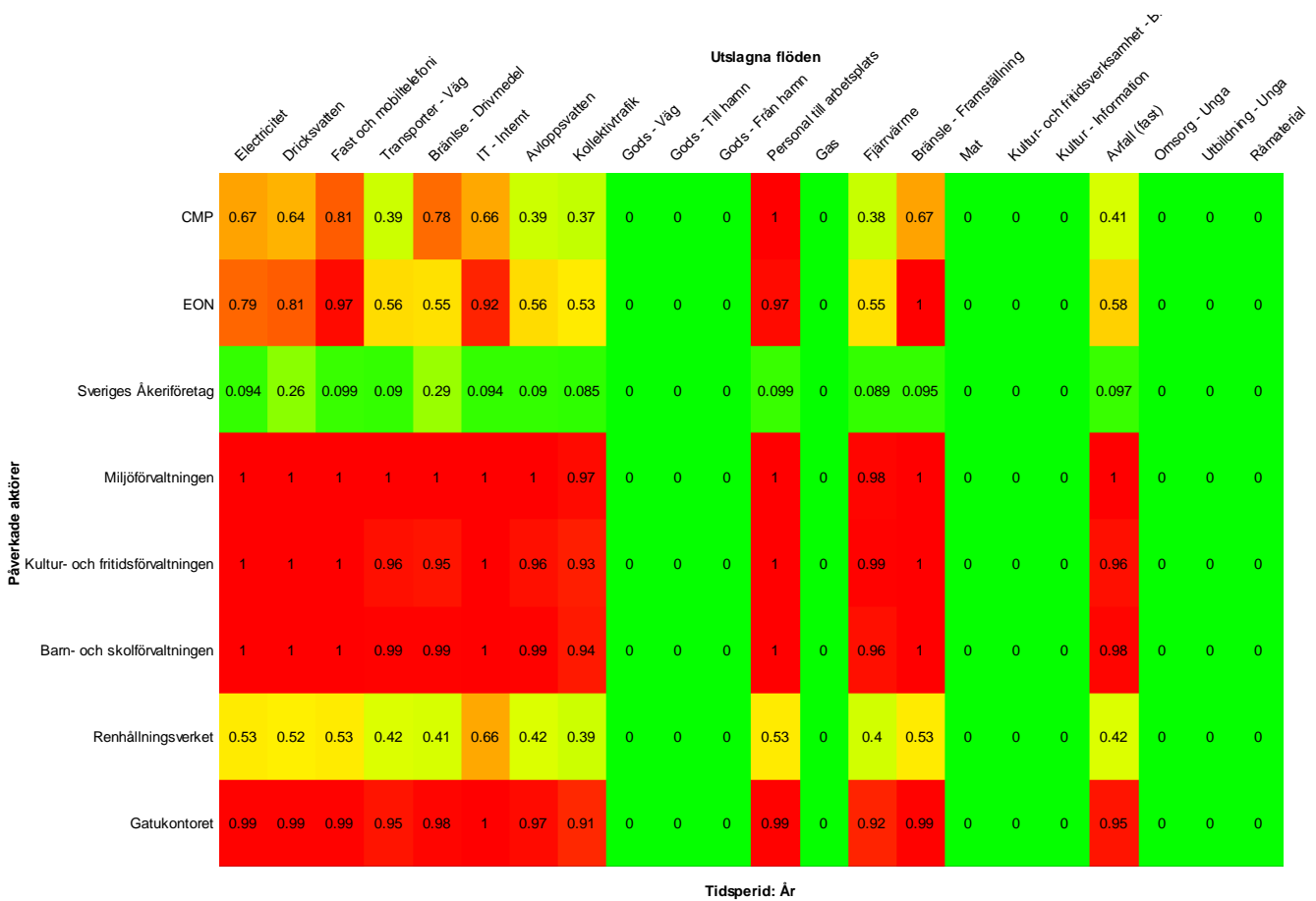
Funktionen presenterar hur beroende aktörer är av flöden under samtliga tidsintervall. Detta görs genom att funktionen skapar en delfigur för varje flöde som påverkar någon aktör. I dessa delfigurer visas aktörens störning när flödet är helt utslaget och hänsyn är taget till fullständig spridning i systemet. Ett exempel med tre flöden, och därmed tre delfigurer, visas i figur 19. Funktionen uppfyller samtliga krav som angivits i kravspecifikationen.



Figur 19. Exempel på aktörers faktiska beroende av flöden för tidsintervall från 0-2 timmar upp till år. 0 innebär full kapacitet och 1 fullständig utslagning av aktören.

6.2.2 Matris över aktörers beroende av flöden

Funktionen skapar en matris som visar hur starkt beroende aktörerna är av respektive flöde för en vald tidsperiod. Beroendet visas genom att respektive flöde slås ut och de påverkade aktörernas kapacitet kan avläsas i figuren, se figur 20. Funktionen uppfyller samtliga krav som angivits i kravspecifikationen.

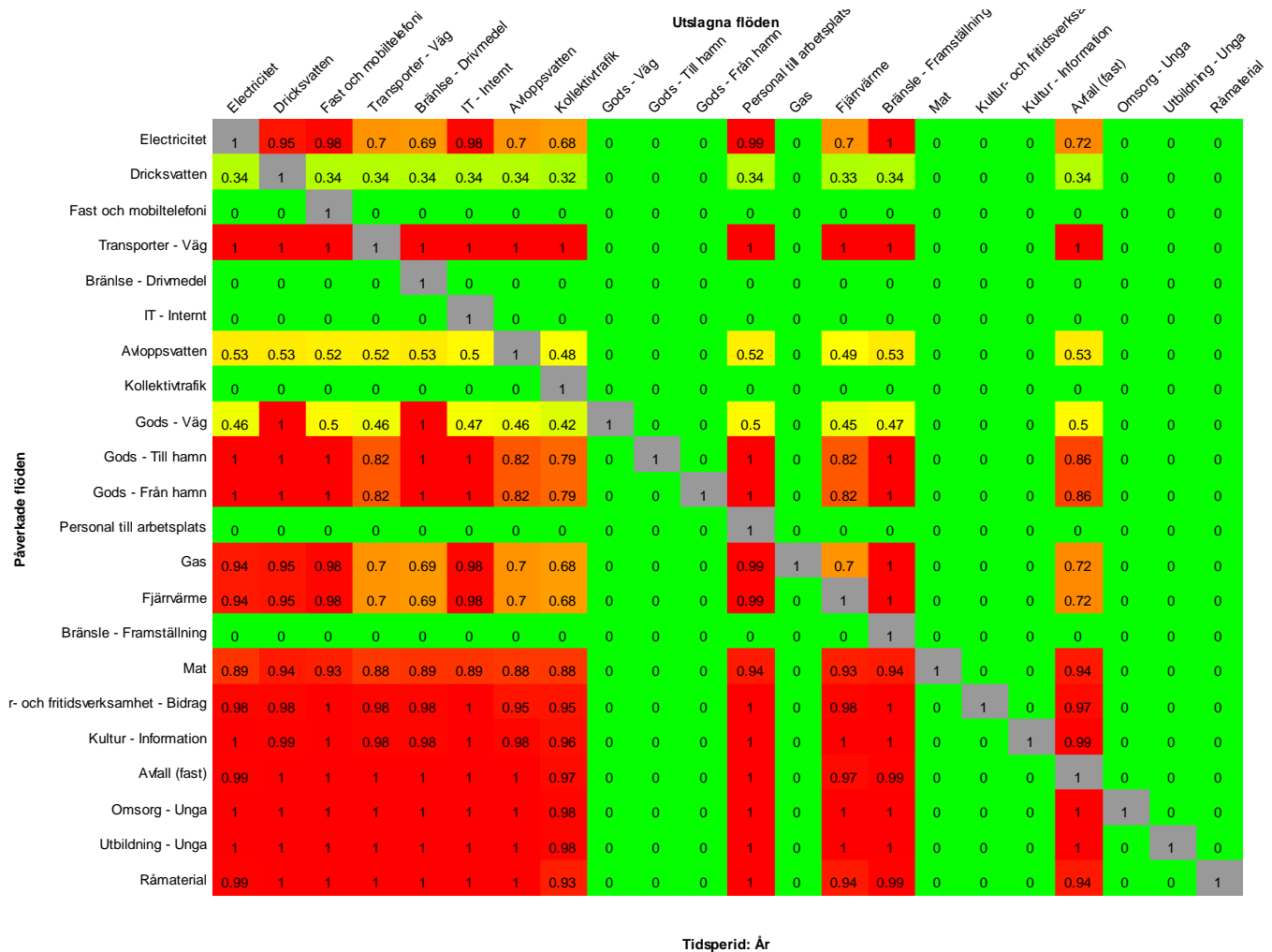


Figur 20. Matrisen visar hur starkt beroende aktörerna är av respektive flöde. Styrkan på beroendet är kapacitetsnedsättningen aktörerna får när respektive flöde slås ut fullständigt. Färgen i matrisen beror på i vilken utsträckning som aktören blir utslagen, röd färg betyder fullständigt utslagen och grön färg betyder opåverkad. Vilken tidsperiod som ska presenteras är valt i datorverktyget.

6.2.3 Matris över flödens beroende av övriga flöden

Funktionen skapar en matris som visar hur starkt beroende flödena är av varandra under en vald tidsperiod, se figur 21. Beroendet visas genom att störningarna hos alla flöden i systemet beräknas när flödena var för sig slås ut helt. Eftersom varje flöde slås ut fullständigt går det inte att beräkna dess

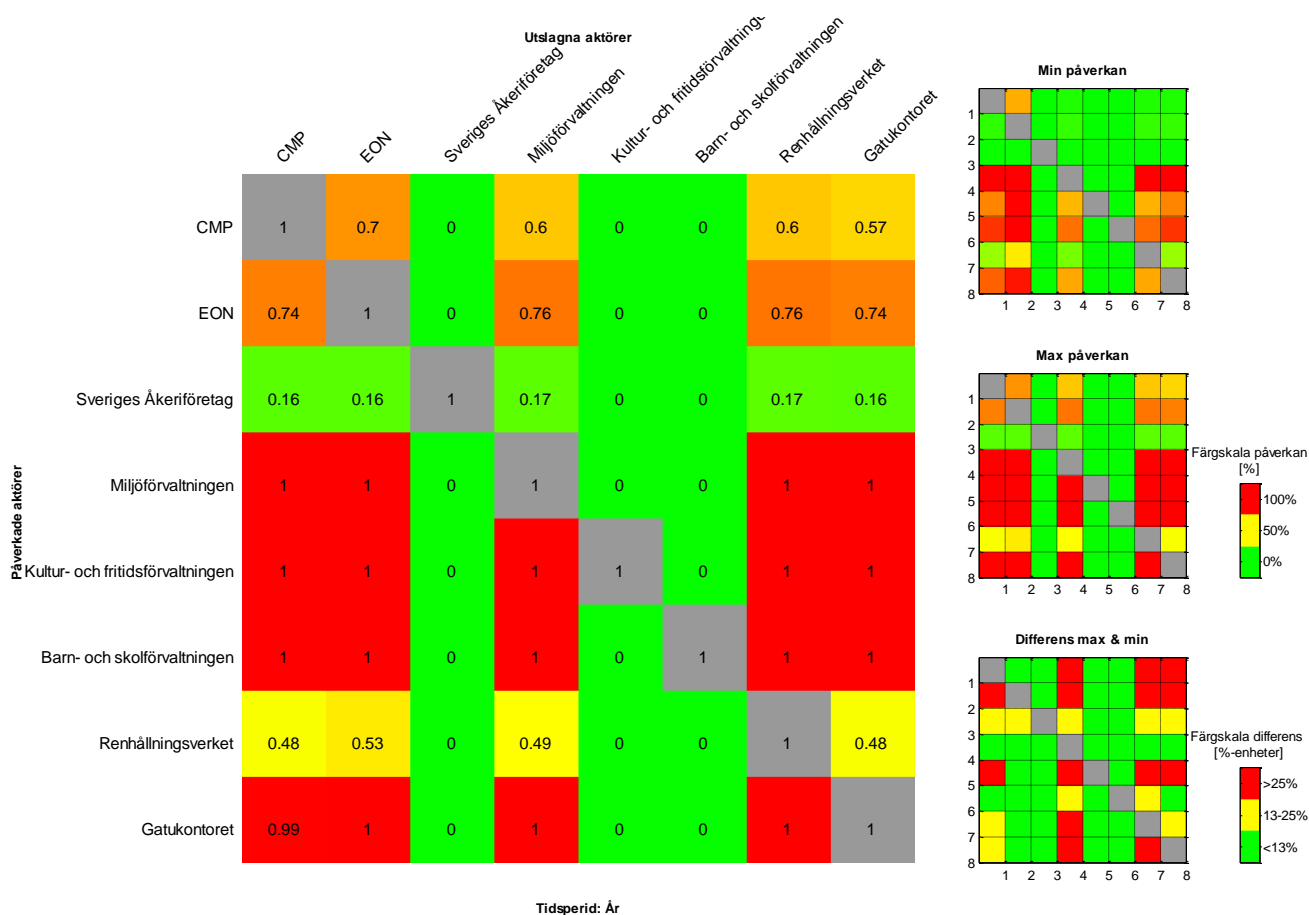
påverkan på sig själv och diagonalen i matrisen beräknas därmed inte. Funktionen uppfyller samtliga krav som angivits i kravspecifikationen.



Figur 21. Matrisen visar hur respektive flöde är beroende av övriga flöden. Styrkan på beroendet är kapacitetsnedsättningen övriga flöden får när respektive flöde slås ut fullständigt. Färgen i matrisen beror på i vilken utsträckning som flödet blir utslagen, röd färg betyder fullständigt utslagen och grön färg betyder opåverkad. Vilken tidsperiod som presenteras är valbart i datorverkyget.

6.2.4 Matris över aktörernas beroende av övriga aktörer

Funktionen skapar en matris som visar hur starkt beroende aktörerna är av varandra, se figur 22. Beroendet visas genom att kapacitetsnedsättningarna hos alla aktörer i systemet beräknas när respektive aktör slås ut. Eftersom varje aktör slås ut fullständigt går det inte att beräkna dess påverkan på sig själv och diagonalen i matrisen, där aktören skulle påverkat sig själv, beräknas därmed inte. Funktionen uppfyller samtliga krav som angivits i kravspecifikationen. Spridningen i resultatet som visas i figurens högra del går att presentera för samtliga funktioner som använder beräkningsmetoden men visas enbart här eftersom det anses tillräckligt även om det vid fortsatt utveckling kan vara lämpligt att presentera i samtliga figurer.



Figur 22. Matris över aktörernas beroende av övriga aktörer inklusive känslighet i resultatet. Vilken tidsperiod som ska presenteras är valbar i datorverktyget. Den stora matrisen visar medelvärdena, den övre av figurerna till höger visar minimal påverkan, figuren i mitten visar maximal påverkan och figuren längst ner visar differensen i minimal och maximal påverkan. För figuren längst ner till höger används en annan färgskala än för övriga figurer, överstiger differensen 25 procentenheter visas rutan som röd, är differensen mindre än 12,5 procentenheter visas den som grön, i övriga fall visas den som gul. 25 procentenheter används som utgångspunkt eftersom det är bredden på de intervall som använts som svarsalternativ vid datainsamlingen.

6.2.5 De viktigaste aktörerna och flödena för valfri aktör och kommun

Funktionen rankar de viktigaste flödena och aktörerna för kommunen och en valfri aktör, se figur 23. Rankningen är baserad på en utslagning av respektive aktör eller flöde med 50 %. Fullständig utslagning används inte eftersom det ska gå att visa aktörer som är beroende av sig själva. 50 % utslagning används eftersom detta enligt responsgrafen, se figur 25, är tillräckligt högt för att ge markant påverkan samtidigt som aktörerna inte slås ut fullständigt. Aktörernas störning räknas samman och normeras till ett intervall mellan 0 till 100 och används för rankningen. När rankningen av aktörerna beräknas dras utslagningen på 50 % bort från resultatet av aktörens påverkan på sig själv. Detta eftersom aktörens beroende av sig själv annars alltid är minst lika stort som utslagningen även om inget faktiskt beroende finns. Rankningen av de viktigaste aktörerna och flödena för kommunen är baserad på summan av alla aktörer i kommunens kapaciteter. Om det samlats in data kring samhällets övergripande mål och hur olika aktörer och flöden inverkar på dessa hade det varit möjligt att ta hänsyn till även detta i en ranking. Detta är något som anses lämpligt och föreslås som fortsatt utveckling. Figuren uppfyller samtliga krav som angivits i

kravspecifikationen.

Kommunens beroende av aktörer		
	Aktör	Beroendefaktor [0-100]
1	EON	68
2	Renhållningsverket	64
3	CMP	63
4	Gatukontoret	60
5	Miljöförvaltningen	58

Kultur- och fritidsförvaltningen beroende av övriga aktörer		
	Aktör	Beroendefaktor [0-100]
1	EON	99
2	Renhållningsverket	95
3	Gatukontoret	95
4	CMP	94
5	Miljöförvaltningen	93

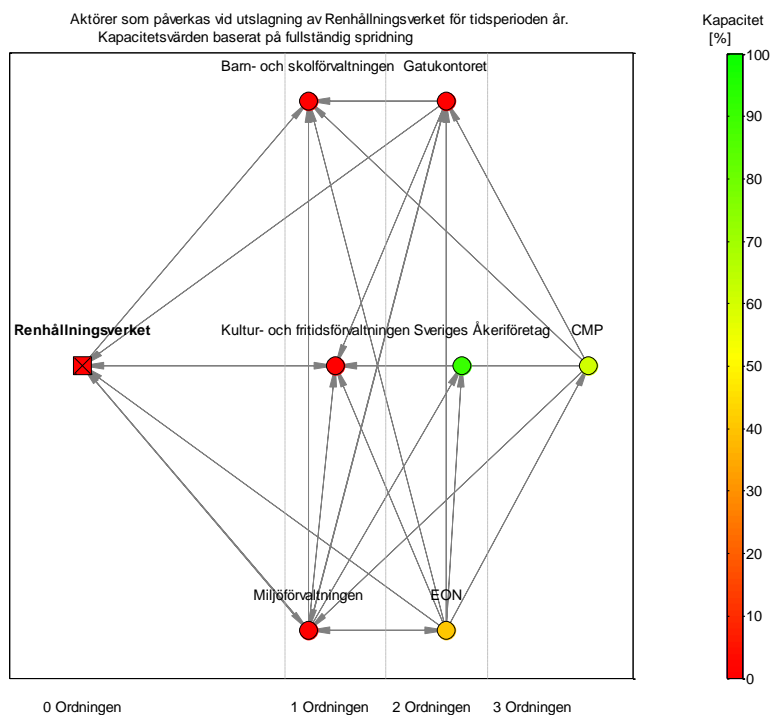
Kommunens beroende av flöden		
	Flöde	Beroendefaktor [0-100]
1	Personal till arbetsplats	28
2	Fast och mobiltelefoni	27
3	IT - Internt	27
4	Dricksvatten	26
5	Bränsle - Drivmedel	26

Kultur- och fritidsförvaltningen beroende av flöden		
	Flöde	Beroendefaktor [0-100]
1	IT - Internt	99
2	Personal till arbetsplats	98
3	Fast och mobiltelefoni	98
4	Bränsle - Framställning	97
5	Electricitet	97

Figur 23. Figuren visar en ranking av de fem mest kritiska aktörerna och flödena för hela kommunen och en enskild valbar aktör. Vilken tidsperiod som ska presenteras är valbart i datorverktyget.

6.2.6 Spridningskedjor från en aktör till alla påverkade aktörer

Funktionen visar vilka aktörer som blir direkt och indirekt påverkade vid utslagning av en vald aktör. Figur 24 visar ett exempel med aktören Renhållningsverket utslagen. Funktionens uppbyggnad presenterades i 6.1.5 men till skillnad från den funktionen beräknas även störningens påverkan på aktörernas kapacitet utifrån analysresultaten för en vald tidsperiod i figur 24. Detta görs genom att det går att ange hur kraftig utslagningen av den första aktören är och därmed kan kapaciteterna för samtliga påverkade aktörer beräknas. Figuren uppfyller samtliga krav som angivits i kravspecifikationen.

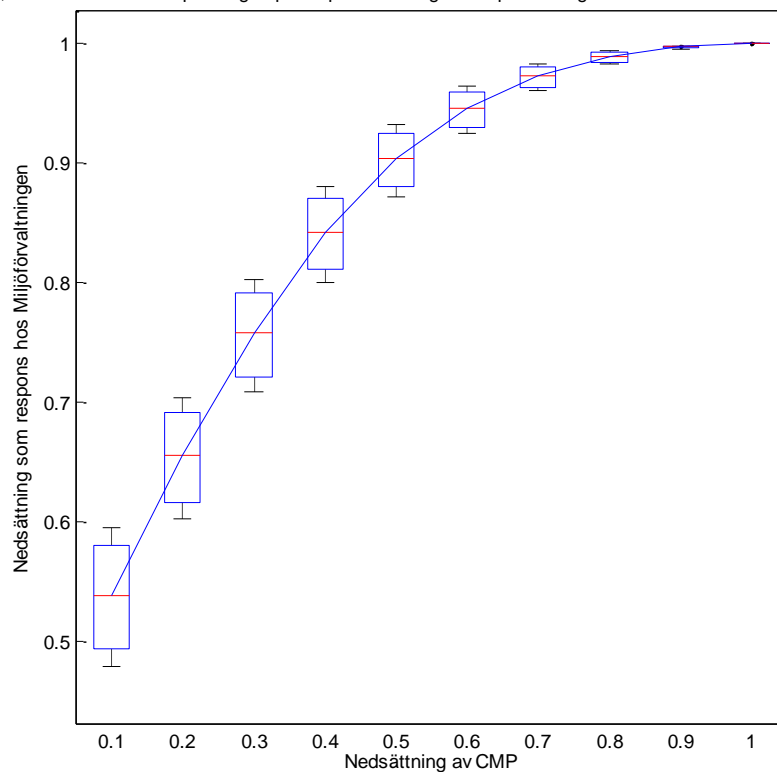


Figur 24. Utslagning av aktör och visualisering av i vilket steg övriga aktörer påverkas och deras kapaciteter. Pilarna går från påverkande till beroende aktör. Vilken tidsperiod som ska presenteras är valbart i datorverktyget.

6.2.7 Responsgraf

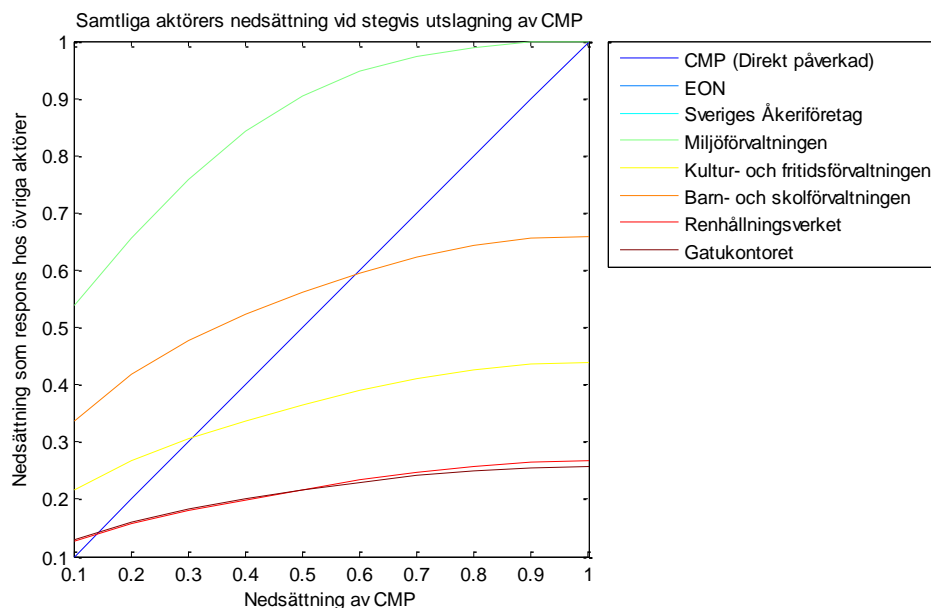
Funktionen skapar två figurer som visar aktörers respons på en störning av en vald aktör. Funktionen har utvecklats framförallt för att analysera valet av beräkningsmetod och omfattas inte av kravspecifikationen. Den första figuren, figur 25, visar responsen hos en aktör när en annan slås ut med 10 % i taget. Figuren visar känsligheten i resultaten genom att presentera låddiagram där beräkningarna är utförda med de lägsta respektive högsta värdena för beroende och påverkan som finns i spridningsintervallet.

Respons hos Miljöförvaltningen vid stegvis utslagning av CMP.
Lådorna visar 25 %, 50 % och 75 % av spridningen på respons när högsta respektive lägsta indatavärde används för beroende och påverkan.



Figur 25. Visualisering av hur stegvis störning hos en CMP påverkar miljöförvaltningen under tidsperioden dygn. Lådorna visar 25 %, 50 % och 75 % av spridningen på respons när högsta respektive lägsta indatavärde används för beroende och påverkan. Morrhåren visar lägsta och högsta respons. Vilken tidsperiod som ska presenteras är valbart i datorverktyget.

Den andra figuren visar responsen för samtliga aktörer när en aktör slås ut stegvis. Ett exempel där aktören CMP har blivit utslagen för tidsperioden dagar syns i figur 26.



Figur 26. Nedsättning hos övriga aktörer som respons på stegvis utslagning av CMP under tidsperioden dygn. Vilken tidsperiod som ska presenteras är valbart i datorverktyget.

6.3 Verifiering

Datorverktyget har systematiskt verifierats med hjälp av strukturerade genomgångar av kodens moduler i enlighet med kapitel 3.3. Metod för verifiering med hjälp av ett testsystem beskrevs i kapitel 3.3 och denna verifiering av datorverktyget presenteras här. Datorverktygets funktioner kontrolleras med ett mindre testsystem som består av tre aktörer och tre flöden. I testfallet har en av de tre aktörerna två aktiviteter och de två övriga aktörerna har varsin aktivitet. För aktören med två aktiviteter finns data för hur aktiviteterna förhåller sig till aktörens mål. Beräkningsmetoden och datorverktyget beräknar varje tidsperiod helt enskilt och därför behöver det bara verifieras för en tidsperiod. Den indata som används till testfallet är därmed konstruerad för en tidsperiod och presenteras i tabell 7, 8 och 9. Aktör 1, aktivitet 1, förkortas till $A_{1,1}$ och Flöde 1 förkortas till F_1 .

Tabell 7. Hur flödet beror av olika aktörer i testfallet.

	$A_{1,1}$	$A_{1,2}$	A_2	A_3
F_1	0,5	0	0,5	0
F_2	0	0,5	0	1
F_3	0,75	0	0	0

Tabell 8. Hur aktörerna beror av flödena i testfallet.

	$A_{1,1}$	$A_{1,2}$	A_2	A_3
F_1	0,75	0,5	0	0
F_2	0	0	0,75	0
F_3	1	0,5	0	0,5

Tabell 9. Hur aktiviteterna för aktör 1 påverkar dess tre mål i testfallet.

	M ₁	M ₂	M ₃
A _{1.1}	4	5	3
A _{1.2}	2	2	2

Vissa av resultaten i datorverktyget kan verifieras utifrån varandra men tre av funktionerna måste verifieras med manuella beräkningar.

- *Matris över aktörers beroende av övriga aktörer*, se figur 22, kan verifiera *responsgrafen* vid en fullständig utslagning, se figur 26. Vilka aktörer som påverkas direkt av en aktörs bortfall och hur mycket de påverkas visas i *spridningskedjan*, se figur 24. Detta kan verifieras med *matris över aktörers beroende av övriga aktörer*. Det är således tillräckligt att verifiera *matris över aktörers beroende av övriga aktörer* för att verifiera dessa tre figurerna, vilket bevisas under kapitel 6.3.1.
- *Matris över aktörers beroende av flöden*, se figur 20, visar aktörernas känslighet vid flödesbortfall under en tidsperiod. Även *aktörers fullständiga beroende av flöden*, se figur 19, visar aktörernas känslighet vid flödesbortfall men utan samma exakthet. Däremot visas störningarna för alla tidsperioder. Det räcker alltså att visa att datormodellen ger korrekta störningar för aktörerna då ett flöde släss ut under en tidsperiod, vilket bevisas under kapitel 6.3.2.
- Figurerna som visar ranking av de viktigaste aktörerna och flödena för en aktör och kommun, se figur 23, och *responsgrafen*, se figur 26, beräknas med hjälp av en viss störning och inte fullständig utslagning. *Responsgrafen* kan användas för att kontrollera att aktörerna i rankingfiguren hamnar i korrekt ordning jämfört med övriga aktörer men den kan inte för att avgöra om värdena är korrekta, då värdena i rankingfiguren är normerade. Att *responsgrafen* är korrekt för 50 % och 100 % utslagning av en aktör bevisas under kapitel 6.3.3.

För att verifiera resultatet från datorverktyget har separata beräkningar genomförts. De separata beräkningarna bygger på ekvationerna som presenteras under kapitel 5.2.2. För testfallet med fyra aktiviteter (två från aktör 1 och en från aktör 2 respektive 3) och tre flöden krävs det 132 matematiska operationer per iteration som beräknas. För att verifiera datorverktygets resultat när störningsspridningen nått stabila värden jämförs resultaten efter tio iterationer. För att automatisera dessa 1320 operationerna per bevis har Microsoft Excel använts. Att Microsoft Excel gör korrekta beräkningar för 2 iterationer har kontrollerats av handberäkningar på papper.

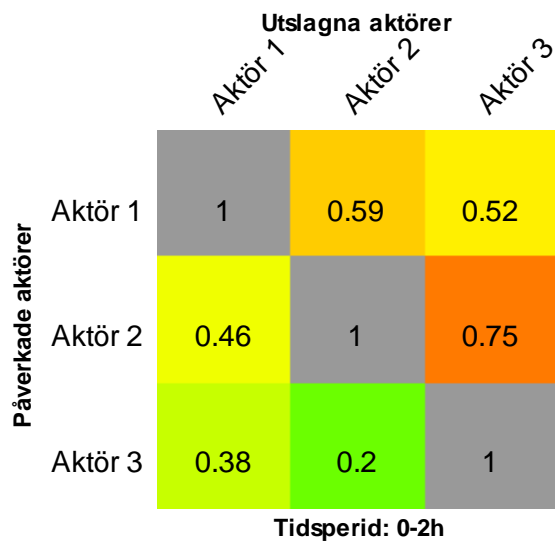
6.3.1 Verifiering av hur aktörer beror av andra aktörer

Verifiering av *matris över aktörers beroende av övriga aktörer* kan genomföras i testfallet genom att studera hur aktör 2 och 3 blir påverkade av en utslagning av aktör 1. Då aktör 1 är utslagen kommer dess aktiviteter att sänka flöde 1, 2 och 3. Aktör 2 och 3 är beroende av flöde 2 respektive flöde 3 och kommer därmed att sänkas av ett bortfall av aktör 1. I tabell 10 presenteras störningen baserat på de manuella beräkningarna efter två och tio iterationer. I tabell 10 presenteras även datorverktygets beräkning av störningen för aktör 2 och 3, vilket står i första kolumnen i figur 27.

Tabell 10. Störningen för aktör 2 och 3 när aktör 1 är utslagen beräknad manuellt och med datorverktyget.

Beräkningsalternativ	Aktör 2 störning	Aktör 3 störning
Manuellt med två iterationer	38 %	38 %
Manuellt med tio iterationer	46 %	38 %
Datorverktyget	46 %	38 %

Den manuella beräkningen med tio iterationer för aktör 2 i tabell 10 visar att störningen är samma som för datorverktyget. Aktör 3 är mindre indirekt beroende av aktör 1 och får därmed ett stabilt värde redan efter den andra iterationen.



Figur 27. Aktörers beroende av andra aktörer i testfallet. Färgskalan i figuren går från grönt till rött där grönt är 0 % påverkan och rött är 100 % påverkan.

Även övergången från aktiviteternas kapacitet till aktörens kapacitet behöver verifieras. Verifieringen av detta sker genom att anta en fullständig utslagning av aktör 3. Aktivitet 1 och 2 för aktör 1 är inte direkt beroende av flöde 2, vilket aktör 3 påverkar, utan de blir istället indirekt beroende av aktör 3 via aktör 2. Viktningen mellan aktivitet 1 och 2 baseras på till hur stor del aktiviteterna uppfyller målen för aktör 1. Med hjälp av värdena i tabell 9 och viktningssantagandet i kapitel 5.1 kommer aktör 1 påverkas med 66,6 % av aktivitet 1 och 33,3 % av aktivitet 2. Förutom de manuella beräkningarna presenteras även störningen enligt datorverktyget i tabell 11 hämtat från första raden, tredje kolumnen, i figur 27.

Tabell 11. Störningen för aktör 1 när aktör 3 är utslagen baserat på datorverktyget och manuella beräkningar.

Beräkningsalternativ	Aktivitet 1 störning	Aktivitet 2 störning	Aktör 1 störning
Manuellt med tre iterationer	28 %	18 %	25 %
Manuellt med tio iterationer	58 %	38 %	52 %
Datorverktyget	58 %	38 %	52 %

Datorverktyget ger samma resultat för störningen av aktör 1 som de manuella beräkningarna med tio iterationer. Med de två ovanstående fallen anses det verifierat att *matris över aktörers beroende av övriga aktörer* stämmer överens med de antagande och ekvationer som presenteras i kapitel 5.

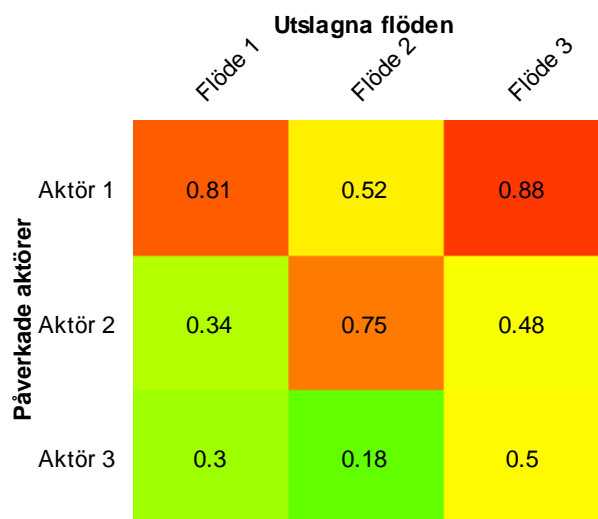
6.3.2 Verifiering av hur aktörer beror av flöden

Verifiering av *matris över aktörers beroende av flöden* kan genomföras i testfallet genom att studera hur de tre aktörernas kapaciteter beror på flöde 1. Enligt tabell 8 är det endast aktör 1 som är beroende av flöde 1 men precis som i föregående kapitel kommer en störningen hos aktör 1 påverka aktör 2 och 3. I tabell 12 presenteras störningen från de manuella beräkningarna efter två och tio iterationer. I tabell 12 presenteras även datorverktygets beräkning av störning för aktör 1, 2 och 3. Resultatet från datorverktyget återfinns också i första kolumnen i figur 28.

Tabell 12. Nedsättningen för aktör 1, 2 och 3 när flöde 1 är utslaget beräknat med datorverktyget och manuellt.

Beräkningsalternativ	Aktör 1 störning	Aktör 2 störning	Aktör 3 störning
Manuellt med två iterationer	80 %	19 %	28 %
Manuellt med tio iterationer	81 %	34 %	30 %
Datorverktyget	81 %	34 %	30 %

Resultatet från de olika beräkningsalternativen skiljer sig inte och därmed är det verifierat att även *matrix över aktörers beroende av flöden* stämmer överens med de antagande och ekvationer som presenteras i kapitel 5.

**Figur 28.** Aktörers beroende av flöden i testfallet. Färgskalan i figuren går från grönt till rött där grönt är 0 % påverkan och rött är 100 % påverkan.

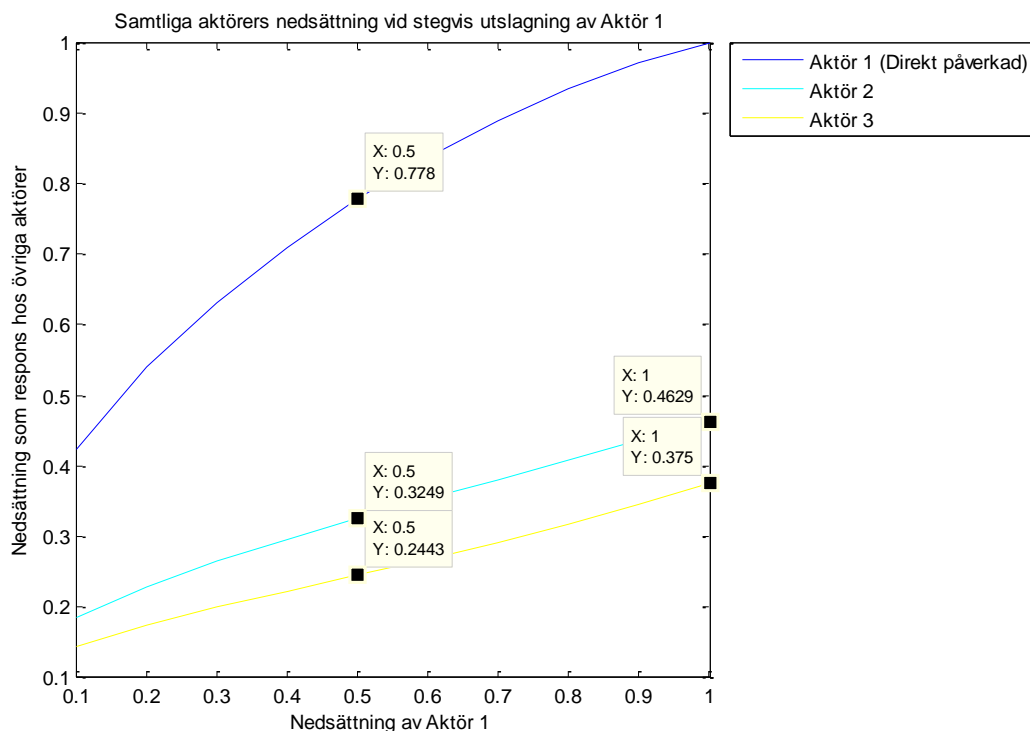
6.3.3 Verifiering av påverkan från aktörer med nedsatt kapacitet

Verifiering av *responsgraf* kan i testfallet genomföras genom att studera hur de tre aktörernas störning påverkas av att aktör 1 initialt har 50 % kapacitet. I tabell 13 presenteras störningen från de manuella beräkningarna efter två och tio iterationer. I tabell 13 presenteras även aktörernas störning vid 0,5 utslagning av aktör 1 i *responsgraf*, figur 29, där aktör 1 har blivit utslaget med 50 %.

Tabell 13. Nedsättningen för aktör 1, 2 och 3 när aktör 1 har 50 % kapacitet beräknat med datorverktyget och manuellt.

Beräkningsalternativ	Aktör 1 störning	Aktör 2 störning	Aktör 3 störning
Manuellt med två iterationer	71 %	19 %	19 %
Manuellt med tio iterationer	78 %	32 %	24 %
Datorverktyget	78 %	32 %	24 %

Värdena i *responsgraf* överensstämmer med de manuella beräkningarna. Noterbart i tabell 13 är även att aktör 1 förutom den initiala störning på 50 % blivit nedsatt med ytterligare 28 % från de flöden som den och övriga aktörer har sänkt.



Figur 29. Hur olika stor initial störning påverkar aktörerna i testfallet.

Responsgraf kan även verifieras genom att störningen för aktör 2 och 3 vid full utslagning av aktör 1 jämförs med figur 27. Verifiering av fullständigt utslagna aktörer gjordes i kapitel 6.3.1 och i första kolumnen i figur 27 är aktör 1 fullständigt utslagen. I tabell 14 presenteras störningen för aktör 2 och 3 med värden från de två olika funktionerna i datorverkyget.

Tabell 14. Kapaciteten för aktör 2 och 3 från olika figurer som visar störningen när aktör 1 är fullständigt utslagen.

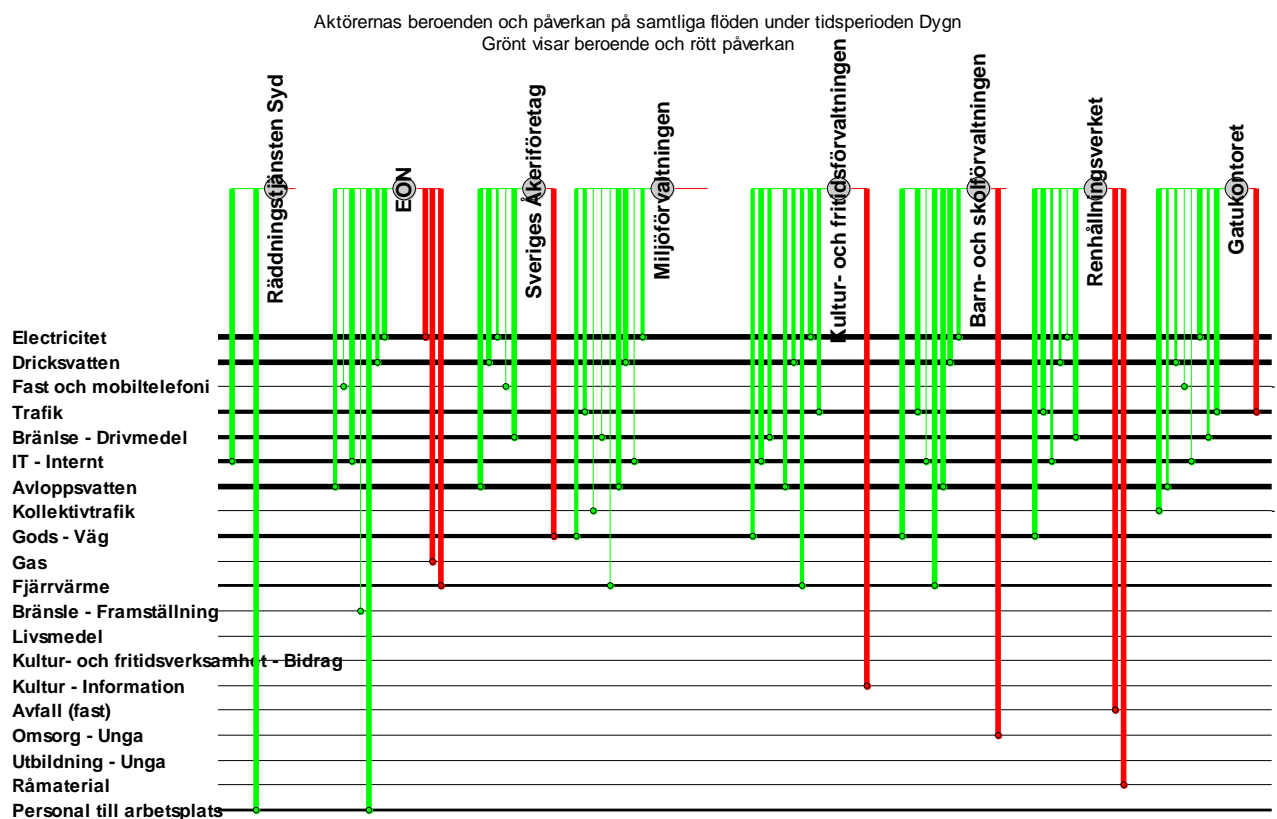
Beräkningsalternativ	Aktör 2 störning	Aktör 3 störning
Responsgraf	46 %	38 %
Matris över aktörers beroende av övriga aktörer	46 %	38 %

Resultatet av tabell 14 visar att de båda funktionerna i datorverkyget ger samma resultat. Då *matris över aktörers beroende av övriga aktörer* verifierades i kapitel 6.3.1 är även resultatet för en fullständig utslagning i *responsgraf* härmed verifierat.

7 Fallstudie – Lunds kommun

Lunds kommun är en av 33 kommuner i Skåne och har drygt 110 000 invånare (SCB, 2013). Storleken på kommunen gör att det finns ett stort antal viktiga aktörer och endast en begränsad del av dessa har blivit kartlagda med hjälp av flödesbegreppet. För att visa på möjligheterna med beroendeanalys ur ett flödesperspektiv och datorverktyget som skapats inom ramarna för detta examensarbete utförs i detta kapitel en fallstudie där beroenden och kritiska aktörer inom Lunds kommun undersöks. Eftersom en beroendeanalys är en del av kommunens risk- och sårbarhetsanalys kan denna fallstudie ses som en utvärdering av datorverktygets användbarhet. Fallstudien baseras på datainsamling som är genomförd av H. Johansson och Åhsberger (2012), J. Johansson och Helmbring (J. Johansson, personlig kommunikation, 16 september, 2013) samt Lindström och Wikman (Under tryckning). Insamlad data är inte heltäckande och data från flera viktiga aktörer saknas. Att indata saknas innebär att flöden som i verkligheten är kritiska blir helt opåverkade i modellen eftersom att ingen aktör som påverkar dem är kartlagd. Vilka flöden som respektive aktör är beroende av och påverkar presenteras i figur 30.

Figur 30 visar vilka flöden varje aktör i kommunen är beroende av och har påverkan på under tidsperioden dygn. Linjerna i figuren visar kopplingarna mellan aktörer och flöden samt deras styrka. Beroenden av flöden visas med gröna linjer. Tjockleken på linjerna bestäms av den av aktörens aktiviteter som är maximalt beroende av flödet. Påverkan på flöden visas med röda linjer och tjockleken bestäms med hjälp av den av aktörens aktiviteter som kan påverkas maximalt. Tjockleken på den svarta linje som går från respektive flöde bestäms av hur många aktörer som är beroende av flödet och styrkan på dessa beroenden.



Figur 30. Samtliga aktörers beroenden och påverkan under tidsperioden dygn.

För att undersöka vilka slutsatser som kan dras från datorverktygets figurer presenteras exempel på samtliga figurer i detta kapitel. Tillsammans med varje figur presenteras möjligheter och begränsningar som framkommit under analysen av Lunds kommun. För att undersöka datorverktygets möjligheter att presentera enskilda aktörer utgår analysen från gatukontoret utöver Lunds kommun i stort. Fallstudien undersöker framförallt hur systemet påverkas av störningar under tidsperioden dygn men även för andra tider i vissa fall. Anledningen till att det framförallt fokuserats på en tidsperiod är att datorverktyget i de flesta figurerna inte tillåter analys av flera tidsperioder samtidigt och hade därmed resulterat i ett oöverskådligt antal figurer.

7.1 Antagande

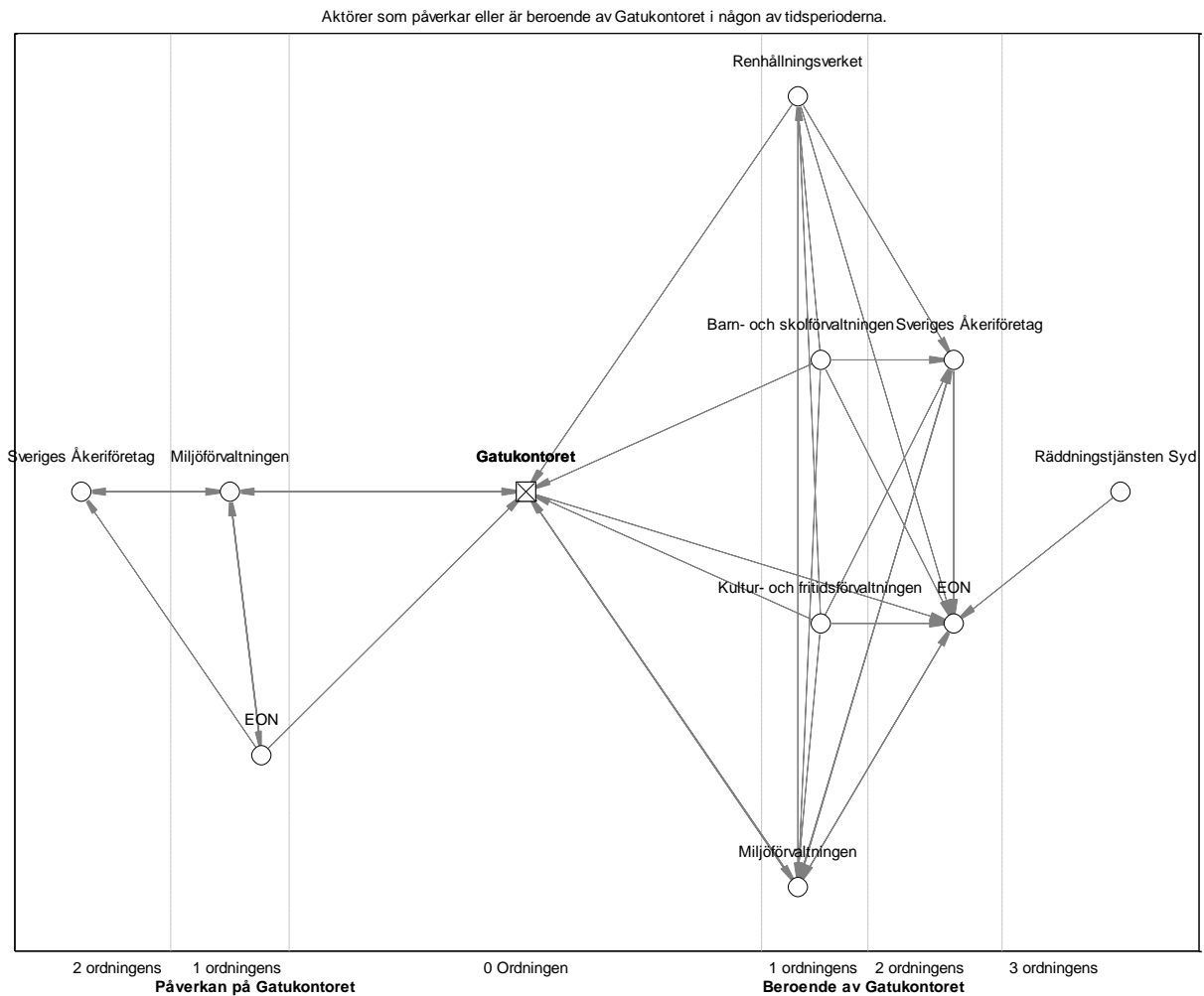
Vissa förändringar har gjorts på tillgänglig data för att den bättre ska överensstämja med Lunds kommun:

- Copenhagen Malmö Port är inte med eftersom den inte bedöms påverka Lunds kommun.
- Data från E.ON och gatukontoret i Malmö används här för att motsvara Krafringen (tidigare Lunds Energi) och gatukontoret i Lund.
- Aktören Sveriges Åkeriföretag representerar ett godtyckligt åkeri med cirka 50 fordon.
- Flödet *person och godstrafik* kan anses inkludera *gods - väg* eftersom det kan vara svåra för aktörer att särskilja dem. För att göra det möjligt för aktörer att vara beroende av antingen godstranporter eller persontransporter på väg har flödet *person och godstrafik* delats upp i de två flödena *gods – väg* och *persontransport - väg*. Många aktörer har angett ett beroende av *person och godstrafik*, med förändringen är de istället beroende av de två nya flödena. De har blivit lika beroende av båda flödena eftersom det inte framgår i vilken omfattning som aktörerna är beroende av persontrafik eller av godstrafik. Aktören gatukontoret som tidigare påverkade flödet *person och godstrafik* har istället antagits påverka flödet *persontransport - väg* och aktören Sveriges Åkeriföretag har antagits påverka flödet *gods – väg*.
- Aktören E.ON har i tillgänglig data angivet ett beroende av flödet *vatten och avlopp*, ett flöde som delats upp i två av övriga aktörer. E.ON har antagits vara lika beroende av flödet *dricksvatten* som av flödet *avloppsvatten*.
- Miljöförvaltningen har angivit en påverkan på flödet *brunnsvatten* men då ingen av aktörerna i fallstudien är beroende av det har flödet inte tagits upp i fallstudien.
- Aktören Räddningstjänsten Syd har inte angivit någon flödespåverkan, vilket leder till ett felmeddelande i datorverktyget. Detta har korrigerats med att Räddningstjänsten Syd påverkar flödet *utbildning – unga* med 0 % för samtliga tidsperioder. Anledningen till att *utbildning – unga* kan användas utan att det leder till ytterligare felaktigheter är att i tillgängliga indata finns det ingen aktör som är beroende av flödet.

7.2 Resultat och inledande analys

Fallstudien introduceras med en kort inledande analys av gatukontoret. För att få en bra överblickbild över gatukontorets roll i kommunen och en uppfattning om vad som kan vara relevant att analysera vidare är figur 31 lämplig eftersom den visar aktörens beroende- och spridningskedjor för samtliga tidsperioder. På bildens vänstra sida visas de tre aktörer som har påverkan på gatukontoret. På den högra sidan syns de aktörer som är beroende av gatukontoret. Figuren visar att ett stort antal aktörer är beroende av gatukontoret och kan det vara lämpligt att undersöka om något av dessa beroenden är mer kritiska än andra. För att undersöka andra aktörers koppling till gatukontoret måste till exempel matrisen över aktörernas beroende av övriga aktörer, figur 37, eller spridningskedjan från gatukontoret, figur 38,

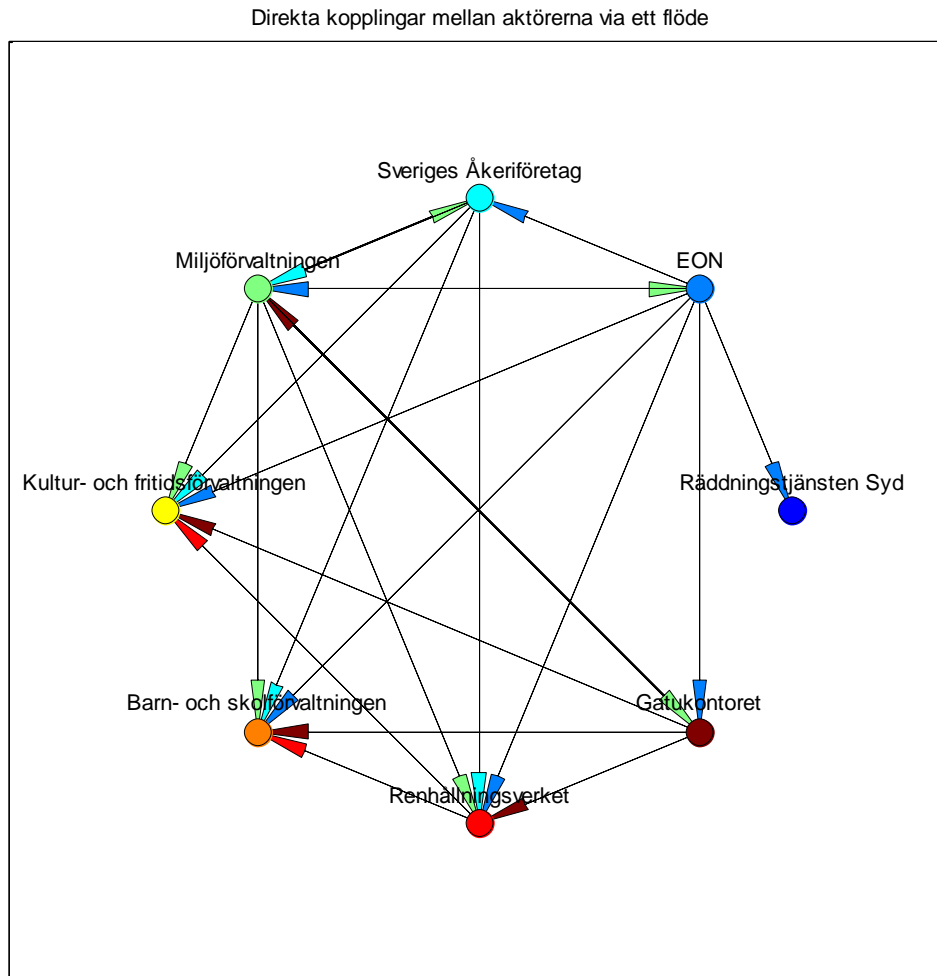
användas då dessa kan ta hänsyn till både olika tidsperioder och styrkor på beroenden vilket inte kan göras i figur 31.



Figur 31. Spridning- och beroendekedjor för gatukontoret under samtliga tidsperioder. Figuren är framtagen direkt från indata utan användning av beräkningsmetoden.

Redan i figur 31 är de aktörer som är beroende av gatukontoret så många att det inte är helt lätt att överskåda, vilket pekar på komplexiteten med beroendeanalyser, och när ett större antal av kommunens aktörer läggs till kommer figuren antagligen att bli oöversiktlig. Det är därmed inte säkert att den är användbar för större system. Genom att till exempel begränsa figuren så att bara upp till tredje ordningens beroende redovisas i denna figur kan den fortsätta vara användbart även för större system.

För att kartlägga kommunens beroenden i stort kan ett första steg vara att titta på aktörernas direkta kopplingar till varandra via flöden, se figur 32, för att se vilka aktörer som direkt påverkar flera aktörer eller är beroende av ett stort antal aktörer. Med hjälp av figuren går det att dra slutsatsen att E.ON påverkar samtliga resterande aktörer i kommunen. Därmed kan det vara lämpligt att fokusera på robustheten hos E.ON. Det går också att dra slutsatsen att både kultur- och fritidsförvaltningen samt barn- och skolförvaltningen är beroende av ett stort antal andra aktörer. Däremot går det inte att säga något om hur starka beroendena är och därför är det svårt att avgöra om det måste analyseras vidare. För att analysera beroendets styrka lämpar sig figuren över aktörernas beroende av övriga aktörer, figur 37, bättre.



Figur 32. Direkta kopplingar mellan aktörer via flöde. Figuren är framtagen direkt från indata utan användning av beräkningsmetoden.

Parallellt med analysen av figur 32 är det lämpligt att använda rankningen av aktörer och flöden, figur 33, eftersom det där går att se hur kritiska olika aktörer och flöden faktiskt är. Från figur 33 utläses att E.ON är en nästan dubbel så kritisk för kommunen som de två näst viktigaste aktörerna Sveriges Åkeriföretag och gatukontoret. De två näst viktigaste aktörerna kan också vara viktiga att analysera vidare. I denna fallstudie fokuseras endast på en aktör, gatukontoret.

För gatukontoret är E.ON är den enda aktören där ett beroende finns under den valda tidsperioden och därmed den enda som undersöks vidare. Genom att skapa rankningen i figur 33 för andra tidsperioder än dygn går det att visa att gatukontoret inte är tydligt beroenden av någon annan aktör än E.ON på tidsperioder upp till månader. Därmed är det för gatukontoret lämpligt att framförallt fokusera på robustheten i kopplingar till E.ON.

Kommunens beroende av aktörer		
	Aktör	Beroendefaktor [0-100]
1	EON	30
2	Sveriges Åkeriföretag	17
3	Gatukontoret	17
4	Renhållningsverket	0
5	Barn- och skolförvaltningen	0

Gatukontoret beroende av övriga aktörer		
	Aktör	Beroendefaktor [0-100]
1	EON	29
2	Renhållningsverket	0
3	Barn- och skolförvaltningen	0
4	Kultur- och fritidsförvaltningen	0
5	Miljöförvaltningen	0

Kommunens beroende av flöden		
	Flöde	Beroendefaktor [0-100]
1	Dricksvatten	15
2	Electricitet	12
3	Avloppsvatten	12
4	Bränsle - Drivmedel	9
5	IT - Internt	9

Gatukontoret beroende av flöden		
	Flöde	Beroendefaktor [0-100]
1	Electricitet	29
2	Avloppsvatten	23
3	Dricksvatten	23
4	Bränsle - Drivmedel	22
5	Trafik	18

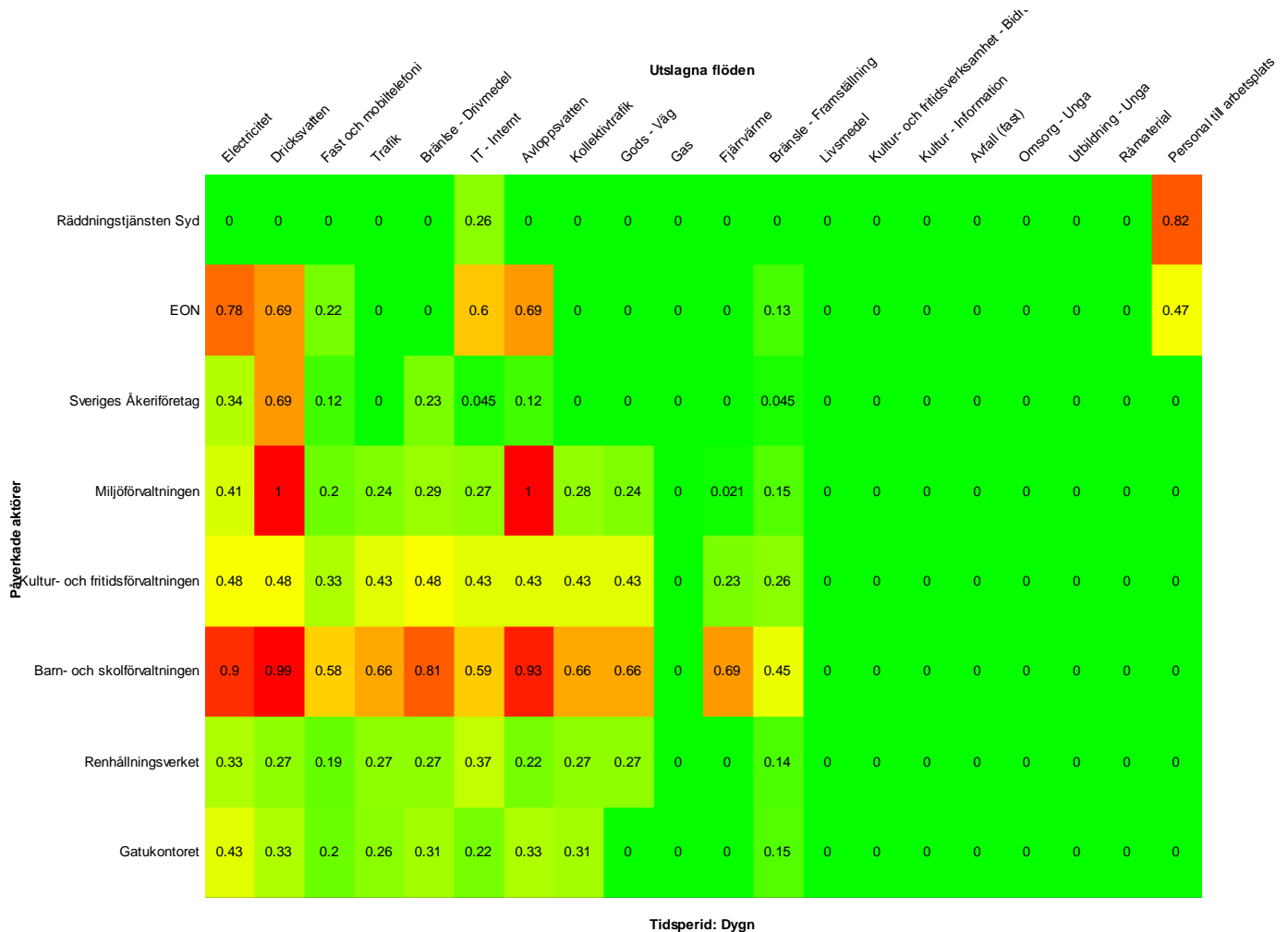
Figur 33. Viktigaste aktörerna och flödena för kommunen och gatuförvaltningen för tidsperioden dygn.

För att förklara varför E.ON får en hög ranking kan till exempel figuren över samtliga aktörers beroenden och påverkan, figur 30, användas för att utläsa vilka flöden som E.ON påverkar. Därefter kan figuren som visar analysresultaten för samtliga aktörers beroende flöden, figur 36, användas för att bestämma hur mycket aktören påverkas av dessa flöden. För att minska gatukontorets utsatthet vid nedsättningar hos E.ON går det att antingen förbättra gatukontorets möjligheter att motstå nedsättningar i flöden som E.ON påverkar eller uppmana E.ON att själva öka sin robusthet. För att öka E.ON:s robusthet och ta reda på vilka av E.ON:s aktiviteter som är mest kritiska skulle figuren som visar E.ON:s direkta flödesberoende för samtliga tidsperioder kunna skapas, motsvarande den analys som genomförts på gatukontoret i figur 35.

För att identifiera de flöden som gör att gatukontoret är beroende av E.ON används figuren över samtliga aktörers beroenden och påverkan, figur 30. I övrigt ger datorverktyget begränsat med möjlighet att beskriva vilka aktörer som påverkar vilka flöden. För tidsperioden dygn påverkar E.ON flödena *fjärrvärme*, *elektricitet* och *gas* men av dessa är det bara *elektricitet* som gatukontoret är direkt beroende av. I matrisen över aktörers beroenden av flöden, figur 34, visas att gatukontorets beroende av *elektricitet* har en styrka på 0,43. Nästan samma styrka, på beroendet av E.ON, 0,46, återfinns senare i figur 37 där beroenden mellan olika aktörer redovisas. Därmed bör det framförallt bara vara flödet *elektricitet* som kopplar ihop aktörerna i tidsperioden dygn.

I matrisen över aktörers beroenden av flöden, figur 34, visas att gatukontoret till exempel skulle påverkas med 0,33 vid en utslagning av flödet *avloppsvatten*. Samtidigt finns det ingen kartlagd aktör som påverkar flödet *avloppsvatten* med i rankningen av aktörer i figur 33, vilket även skulle kunna utläsas i figuren över samtliga aktörers påverkan och beroenden, figur 30. Att flödet finns med i både rankningen i figur 33 och aktörernas beroenden av flöden i figur 34 beror på att dessa visar störningar i flödet oberoende om det, under den valda tidsperioden, påverkas av någon aktör eller inte. Figurerna visar inte heller hur känsliga flöden är för påverkan. Därför kan det till viss del vara vilseledande om figur 34 visar ett starkt beroende av ett flöde eftersom flödet i sig kan vara robust. Samtidigt betyder det starka beroendet att aktören kanske bör analysera de stötdämpare som är kopplade till flödet även om någon aktör som stör flödet inte är identifierad. Detta problem uppstår i huvudsak när inte tillräckligt många aktörer är kartlagda. För vidare diskussioner kring hur kartläggningen ska styras rekommenderas Lindström och Wikman (Under tryckning).

I matrisen över aktörernas beroende av flöden, figur 34, går det att se att det flöde som är viktigast för kommunen är *dricksvatten* men såväl *elektricitet* som *avloppsvatten* är viktiga att studera. När det är säkerställt vilka aktörer som kan vara mest kritiska för upprätthållandet av dessa flöden kan de aktörerna analyseras vidare med till exempel figuren över en aktörs direkta flödesberoende och påverkan, motsvarande den för gatukontoret i figur 35.

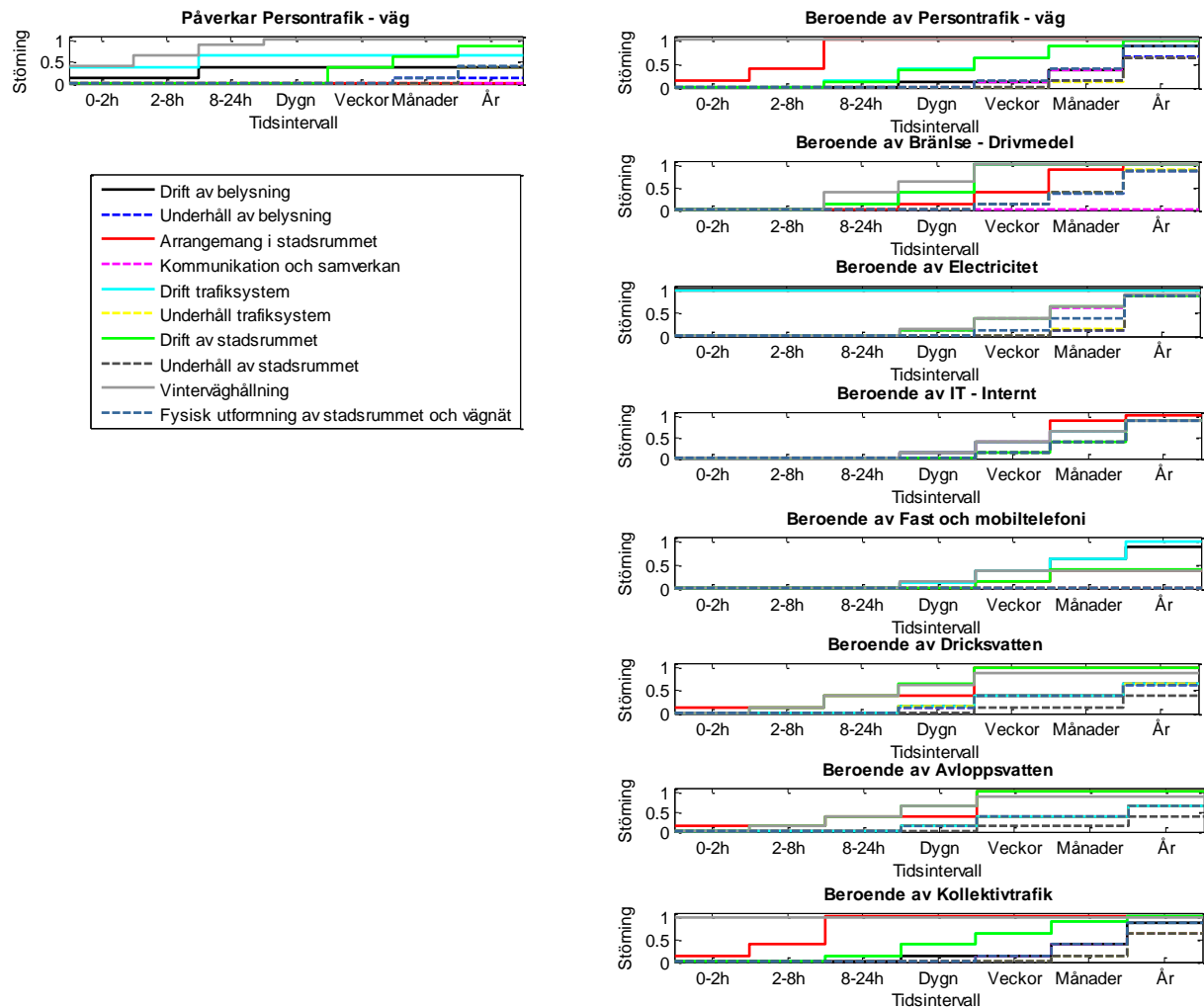


Figur 34. Matris över aktörers beroende av flöden för tidsperioden dygn.

Rankningen i figur 33 med de viktigaste flödena för en enskild aktör kan användas framförallt som ett stöd för att öka aktörens robusthet. I rankningen är *elektricitet* högt rankat för gatukontoret och genom att granska figur 35 går det att avgöra vilka av gatukontorets aktiviteter som har starkast beroende till *elektricitet*.

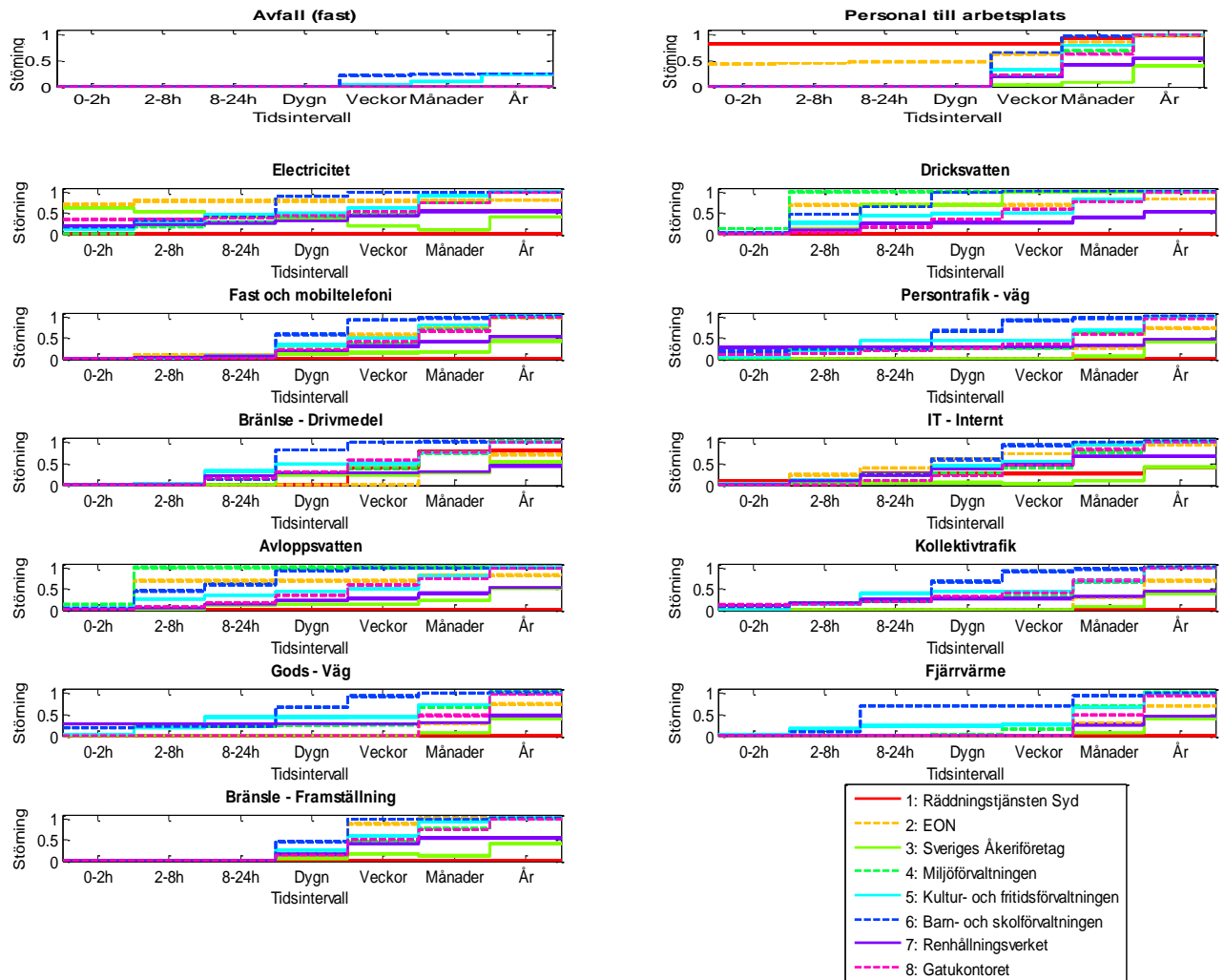
Även analysen som genomfördes under figur 33 ledde till slutsatsen att flödet *elektricitet* var mest kritiskt för gatukontoret. I figur 35 går det att koppla detta beroende till att aktiviteterna *Drift trafiksystem* och *Arrangemang i stadsrummet* blir fullständigt utslagna vid ett bortfall av flödet *elektricitet*. I figuren går det att i detalj studera gatuförvaltningen och då går det att komma fram till att det är aktiviteten *Drift trafiksystem* som är mest avgörande för flödet *persontransport - väg* bortsett från *vinterväghållning* som är en säsongsbetonad aktivitet. Säsongsbetonade aktiviteter som *Vinterväghållning* diskuteras vidare i kapitel 7.4.

Flödespåverkan och flödesberoende för Gatukontoret



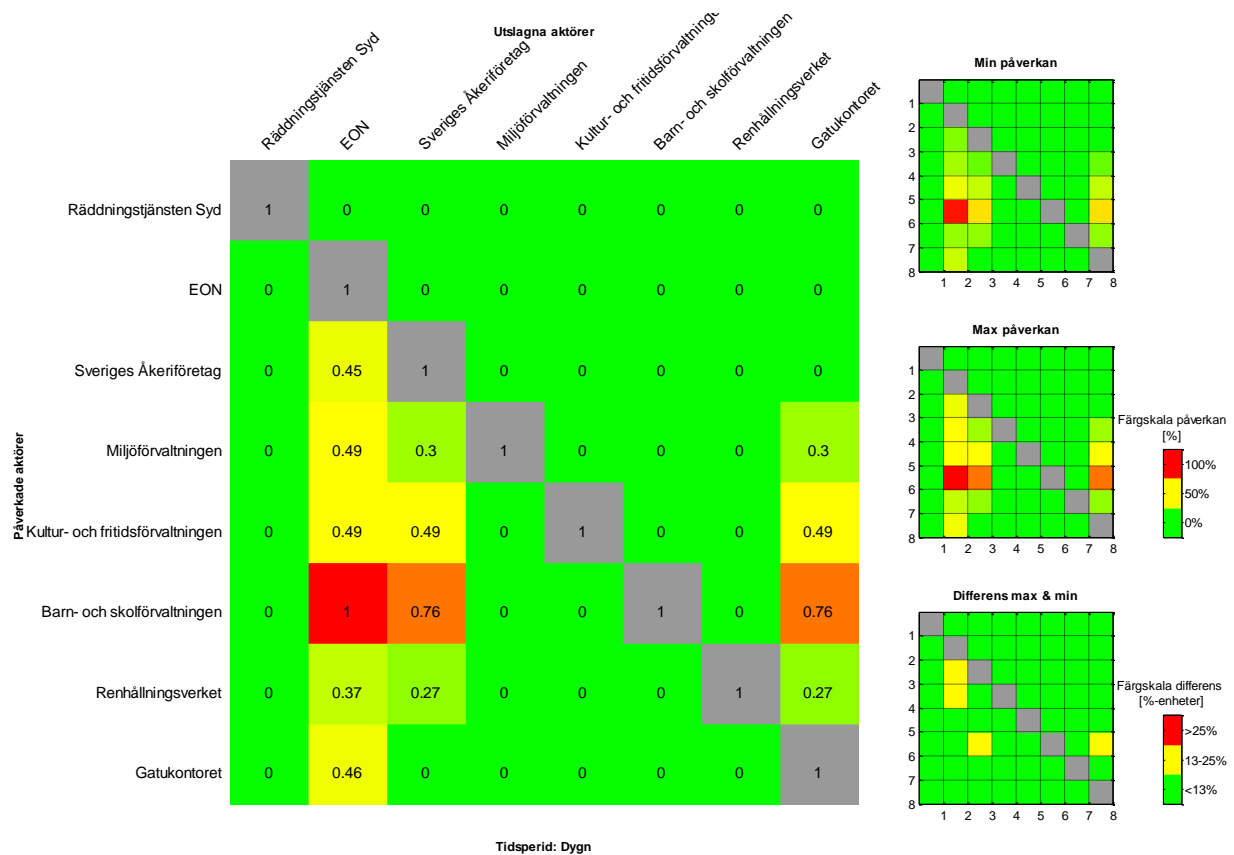
Figur 35. Gatukontorets direkta flödespåverkan och flödesberoende för samtliga tidsperioder. Figuren är framtagen direkt från indata utan användning av beräkningsmetoden.

Med hjälp av figur 36 går det att snabbt få en översikt över när flöden blir kritiska även om det är svårt att läsa av exakta störningar. Ur kommunens perspektiv kan det vara intressant att framförallt *elektricitet* och *dricksvatten* snabbt blir kritiskt för flera aktörer men även att *persontransport - väg* och *avloppsvatten* är viktiga, vilket inte behöver vara lika enkelt att förutsäga. Figur 36 är även ett bra hjälpmedel till analysen efter figur 33 eftersom att det går att se hur aktörernas beroende ser ut för andra tidsperioder än dygn. Till exempel kan figur 36 visa gatukontoret att flödet *persontransport - väg* inte påverkar någon aktör mer än 0,5 förrän än efter ett dygn. Detta skulle kunna hjälpa gatukontoret att prioritera sina resurser.



Figur 36. Analysresultatet för samtliga aktörers beroende av flöden under samtliga tidsperioder.

Med hjälp av matriserna i figur 34 och 37 som visar aktörernas beroende av flöden och andra aktörer går det att dra samma slutsatser som med rankningen i figur 33. Fördelen med att gå via figur 33 kan vara att det är lättare att snabbt få en överblick, över vilka aktörer och flöden som är viktigast för både kommunen och aktören. Ur kommunens perspektiv kan det vara bra att uppmärksammas på att till exempel barn- och skolförvaltningen är starkt beroende av ett stort antal flöden, något som också återspeglas i figur 37 där flera aktörer har stark påverkan på förvaltningen. Därför kan det vara relevant för kommunen att vidare analysera möjligheten att införa stötdämpare för barn- och skolförvaltningen.

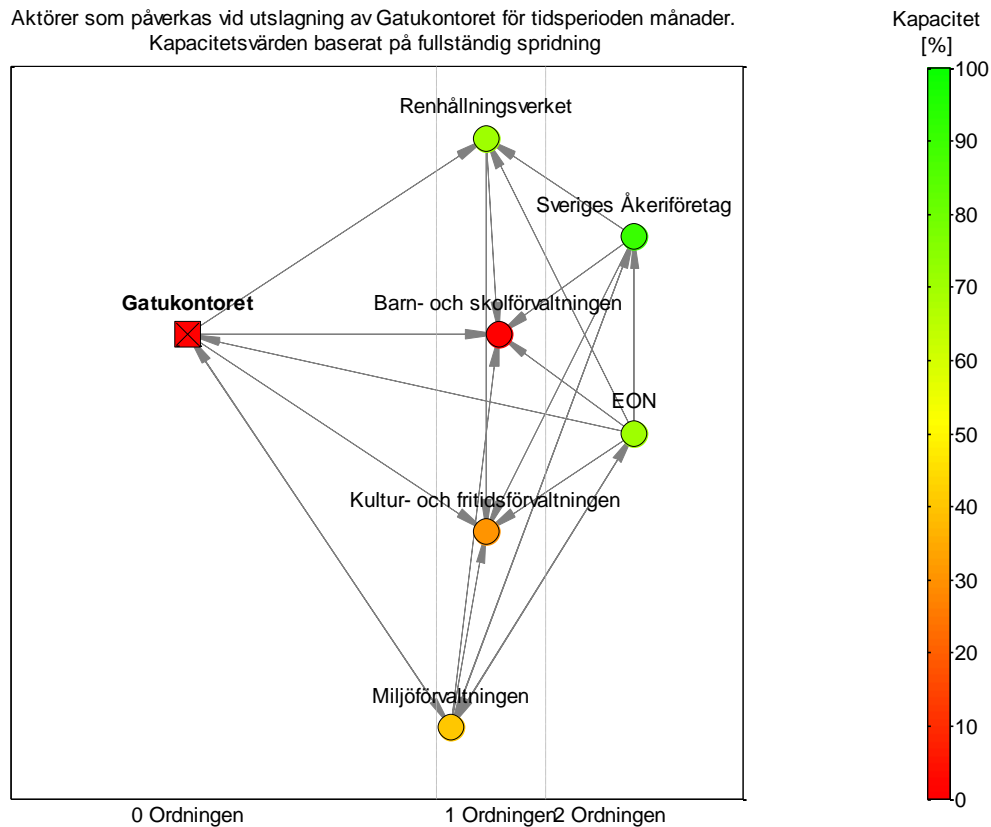


Figur 37. Matris över aktörernas beroende av övriga aktörer inklusive känslighet i resultatet för tidsperioden dygn. Den stora matrisen visar medelvärdena, den övre av figurerna till höger visar minimal påverkan, figuren i mitten visar maximal påverkan och figuren längst ner visar differensen i minimal och maximal påverkan. För figuren längst ner till höger används en annan färgskala än för övriga figurer, överstiger differensen 25 procentenheter visas rutan som röd, är differensen mindre än 12,5 procentenheter visas den som grön, i övriga fall visas den som gul. 25 procentenheter används som utgångspunkt eftersom det är bredden på de intervall som använts som svarsalternativ vid datainsamlingen.

Ur kommunens perspektiv visar figurerna ovan att E.ON snabbt påverkar många aktörer men att även gatukontoret och Sveriges åkeriföretag påverkar många aktörer. Spridningen i påverkan och beroende som går att avläsa i figurerna till höger är viktigt att kontrollera innan slutsatser dras från figurerna. I det här fallet är E.ON aktören med mest påverkan även vid minsta och högsta värden vilket gör att den bör fortsätta vara mest intressant. För gatukontoret visar samtliga figurer att beroende bara finns till E.ON men inte via vilka flöden det gäller. Vilket flöde det gäller har tidigare visats med hjälp av figuren som visar samtliga aktörers beroenden av och påverkan på flöden, figur 30. I figuren är det enbart flödet elektricitet som kopplar samman aktörerna och styrkan på beroenden kan studeras i figuren som visar samtliga aktörers beroende av flöden, figur 36 vilket även detta redovisats tidigare. Eftersom figur 34 och 37 inte ger samma beroende mellan gatukontoret och E.ON som *elektricitet* finns där indirekta beroende som gör att E.ON är viktigare för gatukontoret än flödet *elektricitet*. För att förenkla analysen av aktörers kopplingar till varandra hade det varit önskvärt att skapa en figur som kopplar ihop aktören med de aktiviteter som orsakar påverkan och beroenden.

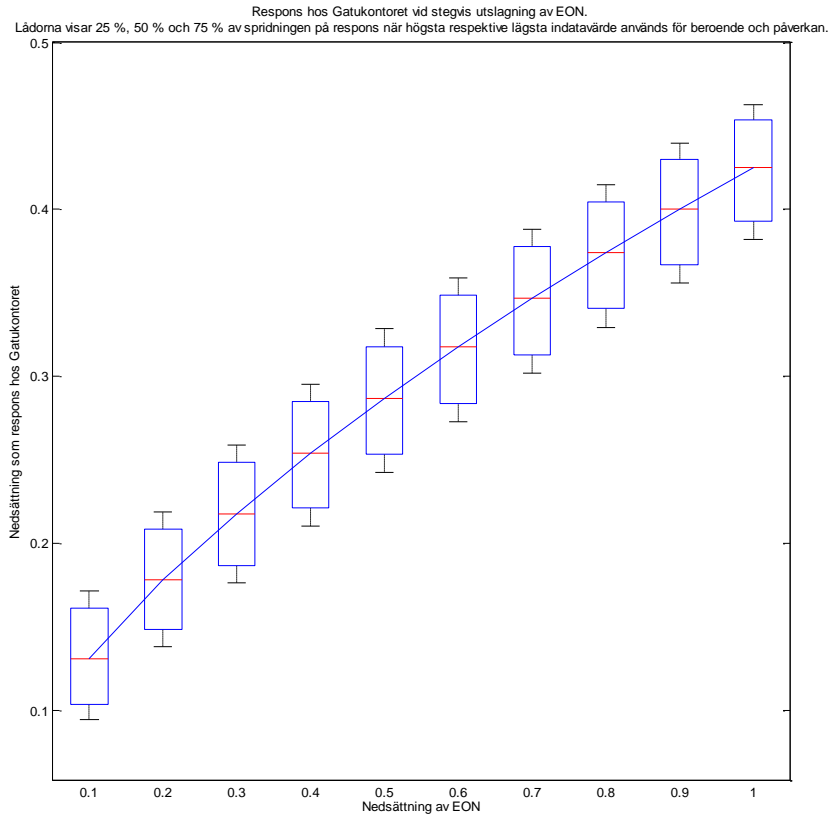
För att undersöka hur olika kraftiga störningar av gatukontoret påverkar övriga aktörer kan spridningskedjorna i figur 38 användas. Ur figuren kan det utläsas att barn- och skolförvaltningen blir kraftigast påverkad vid en 100 % utslagning av gatukontoret. Ur kommunens perspektiv kan det vara relevant att visa vilka aktörer som påverkas vid en utslagning. För gatukontoret bör det vara mer intressant att undersöka versioner av figur 38 där det istället syns hur gatukontoret påverkas vid utslagning av andra aktörer. Ett exempel på detta är om figur 38 används av barn- och skolförvaltningen för att visa gatukontoret vilken betydelse de har. Spridningskedjorna i figur 38 liknar de beroende- och

spridningskedjor som återfinns i figur 31 men med skillnaden att figur 38 visar hur mycket som andra aktörer blir påverkade av störningar hos gatukontoret.



Figur 38. Spridningskedja från gatukontoret till alla påverkade aktörer.

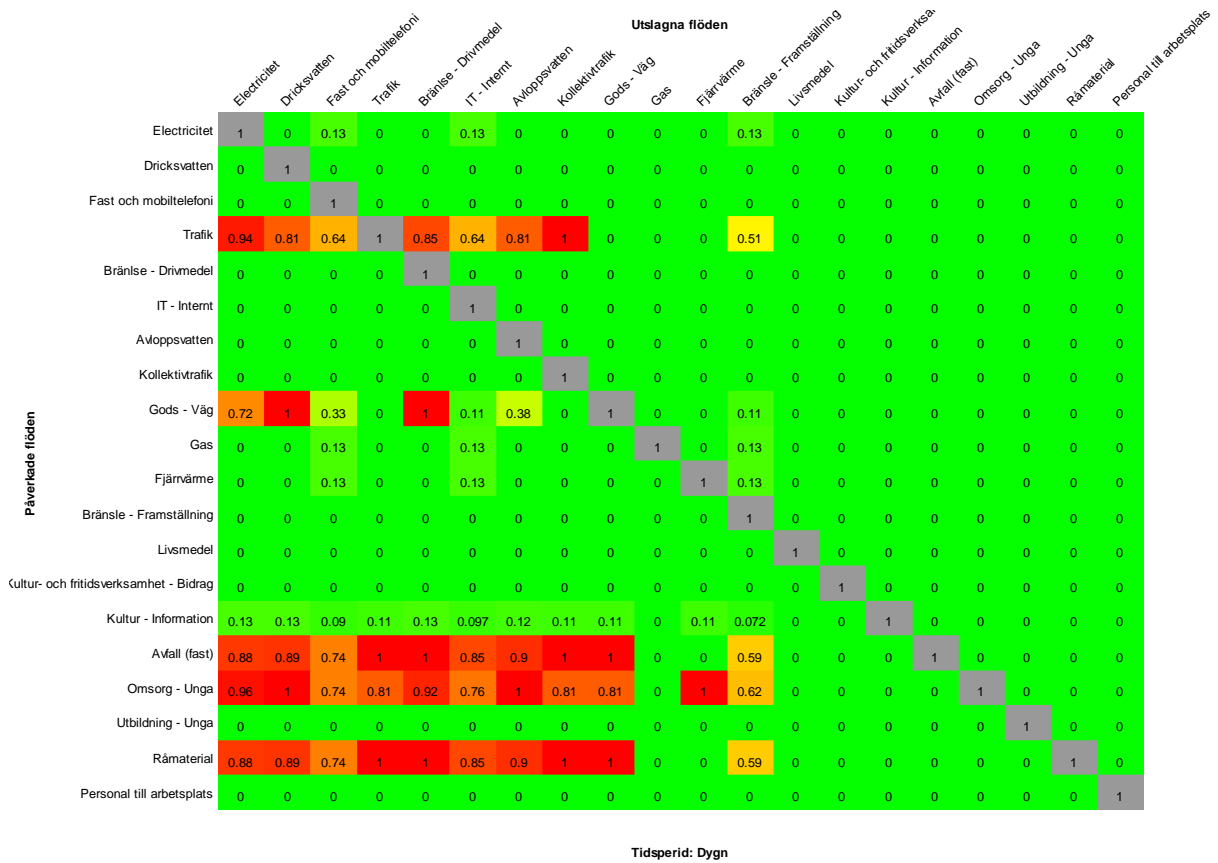
För att granska hur andra störningar av gatukontoret påverkar barn- och skolförvaltningen kan fler versioner av spridningskedjorna i figur 38 genereras alternativt en version av responsgraden i figur 39 som istället visar utslagning av gatukontoret och responsen hos barn och skolförvaltningen. Båda figurerna kan dessutom visa hur förbättringar i gatukontorets robusthet leder till lägre störningar hos de beroende aktörerna. För spridningskedjan i figur 38 görs detta genom att en lägre störning matas in och för responsgraden i figur 39 avläses istället värdet för störningen för ett annat värde på x-axeln.



Figur 39. Responsgraf som visar respons hos gatukontoret vid stegvis utslagning av E.ON. Lådorna visar 25 %, 50 % och 75 % av spridningen på respons när högsta respektive lägsta indatavärde används för beroende och påverkan. Morrhåren visar lägsta och högsta respons.

För att identifiera vilka flöden som behöver undersökas vidare och var mer data måste samlas in kan det vara lämpligt att använda figur 40. De flöden som helt saknar påverkan under samtliga tidsperioder är exempel på sådana som antagligen kräver mer datainsamling. I figur 40 finns ett stort antal flöden utan beroenden eller påverkan, vilket visar på behovet av ytterligare datainsamling innan en fullständig fallstudie kan utföras för Lunds kommun. Figuren möjliggör också en överblick över vilka flöden som påverkar varandra och hur en störning sprider sig, däremot är det svårt att dra slutsatser eftersom kopplingen till aktörer måste göras manuellt. Denna överblick kan vara intressant för exempelvis gatukontoret som bör ha ett övergripande intresse av att flödet *persontransport - väg* upprätthålls även om ett stort antal aktörer antagligen påverkar flödet i ett verkligt system.

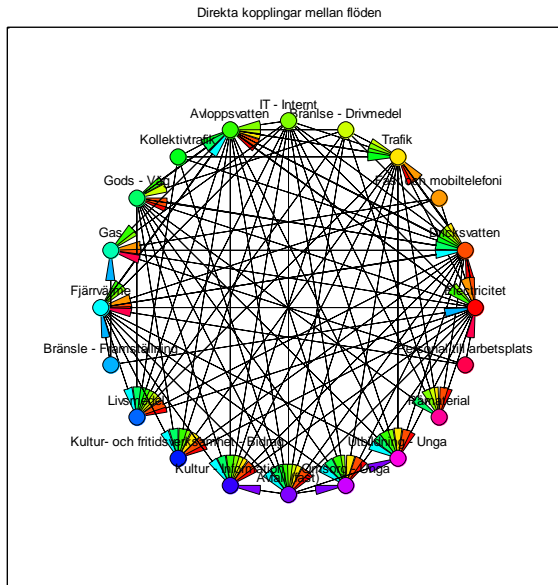
7.3. Figurer som skapats men inte använts i fallstudien



Figur 40. Matris över flödens beroende av övriga flöden.

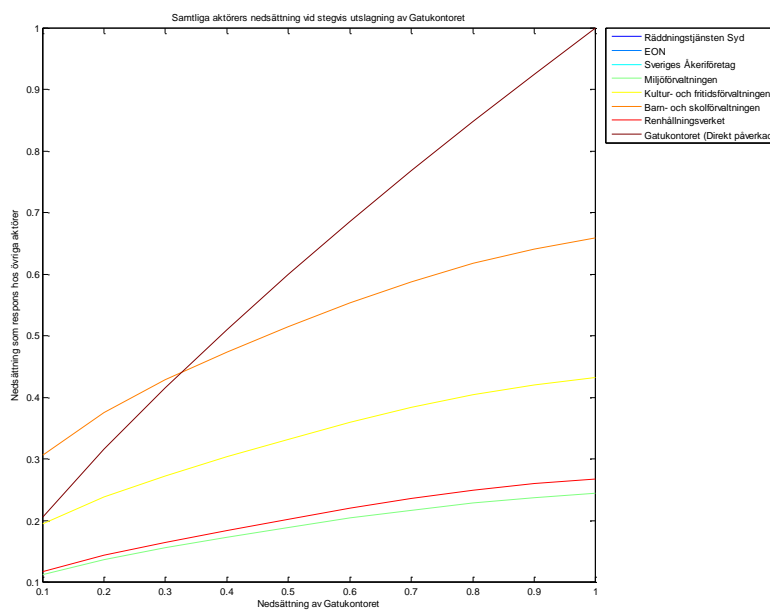
7.3 Figurer som skapats men inte använts i fallstudien

Två av figurerna som kan skapas med datorverktyget har inte varit till användning i fallstudien och presenteras därför här. Generellt är det flera av figurerna datorverktyget skapar som är redundanta och visar ungefär samma sak men på olika sätt. Figur 41 anses inte vara användbar utöver att den visar komplexiteten i systemet. Detta då den även med ett förhållandevis begränsat antal flöden blir oöverskådlig.



Figur 41. Direkta kopplingar mellan flöden. Figuren är framtagen med direkt från indata utan användning av beräkningsmetoden.

Figur 42 medger inget enkelt sätt för användaren att dra slutsatser och har därför inte använts i fallstudien. Däremot kan den vara användbar för en mer insatt användare för att undersöka vad olika störningar ger för respons hos det resterande systemet. Figuren är framförallt skapad för att visualisera modellen och inte för att användaren ska dra slutsatser.



Figur 42. Respons för hos hela systemet vid störning av aktören gatukontoret för tidsperioden dygn.

7.4 Diskussion – Fallstudie

Generellt saknas det för många aktörer och flöden för att datorverktyget ska vara till hjälp vid en beroendeanalys för Lunds kommun. Det som kommer diskuteras nedan är de antagande som är gjorda samt några slutsatser som kan vara användbara i den beroendeanalys som är en del av kommunens risk- och sårbarhetsanalys.

7.4.1 Diskussion kring antaganden

Konsekvensen av att dela upp flödet *person och godstrafik* på *gods – väg* och *persontransport - väg*, enligt antagandet i kapitel 5.1, är att fler aktörer har fått ett beroende av Sveriges åkeriföretag. Eftersom gatukontoret är beroende av *gods – väg* kommer antagandet, av ett extra beroende, ge Sveriges åkeriföretag en förhöjd påverkan på övriga aktörer.

Konsekvensen av att dela upp E.ON:s beroende av *vatten och avlopp* till två separata flöden, *dricksvatten* och *avloppsvatten*, är att E.ON:s beroende av miljöförvaltningen ökar. Detta bedöms vara rimligt eftersom detta beroende gäller även för övriga aktörer som behöver dricksvatten.

Eftersom en brist i datorverktyget gör att det krävs att alla aktörer påverkar något flöde förändrades Räddningstjänsten Syds indata till att 0 % påverkan på flödet *utbildning – unga*. Förändringen ger inga effekter på resultatet utöver att datorverktyget kan köras.

För att använda resultatet från datorverktyget för en fullständig risk- och sårbarhetsanalys för Lunds kommun rekommenderas det att ovanstående antagande ersätts med verklig indata.

7.4.2 Övrig diskussion av fallstudien

I datorverktyget är det svårt att dra slutsatser som gäller över flera tidsperioder eller hur beroende och påverkan förändras över tiden. Det hade till exempel varit önskvärt med en animation för spridningskedjor eller att se hur matrisen för aktörers beroende av aktörer förändras med tiden. Detta är också något som föreslås som fortsatt arbete i kapitel 9.1.1.

I datorverktyget presenteras kopplingen från aktör till flöde och flöde till aktör endast i figur 30, vilken kan vara svår att ta till sig. Det hade varit önskvärt att datorverktyget i till exempel spridningskedjan kunde skriva ut vilka flöden som aktörer påverkar varandra med.

Gatukontoret har angivit att aktiviteten *vinterväghållning* påverkar flödet *persontransport - väg* kraftigt och redan efter ett dygn slås flödet ut fullständigt, se figur 35. Aktiviteten *vinterväghållning* är endast aktuell under vissa delar av året vilket gör påverkan från gatukontoret något missvisande, vilket diskuteras vidare i kapitel 8.3. Gatukontoret utför dock andra aktiviteter som påverkar flödet *persontransport - väg* vilket gör att resultatet bara är missvisande för den extra delen av påverkan som *vinterväghållning* står för. Att Räddningstjänsten Syd inte har någon flödespåverkan anses inte helt troligt. Räddningstjänsten Syd bör påverka något flöde när de till exempel genomför aktiviteterna *extern utbildning* och *skadeavhjälpanse insatser*.

Då figur 34 analyseras framgår det att bland annat barn- och skolförvaltningen är beroende av flödet *bränsle – framställning* med 45 %. Detta trots att barn- och skolförvaltningen saknar ett direkt beroende till flödet. Barn- och skolförvaltningen är däremot beroende av flödena *elektricitet* och *fjärrvärme*, flöden som E.ON påverkar. E.ON är beroende av flödet *bränsle – framställning* för att inte påverka flödena *elektricitet* och *fjärrvärme*. Det som är intressant med barn- och skolförvaltningens beroende av flödet *bränsle – framställning* är att barn- och skolförvaltningen har ett drygt tre gånger större beroende av flödet än vad E.ON har. Förklaringen till detta ligger i hur antagandet om hur aktivitetens kapacitet påverkar aktörers kapacitet. Barn och skolförvaltningens viktigaste aktiviteter, baserat på deras mål, är starkt beroende av E.ON:s flöden. Detta medför att en liten sänkning i E.ON:s kapacitet sänker barn- och skolförvaltningens kapacitet kraftigt. Flertalet av E.ON:s aktiviteter är även beroende av *fjärrvärme* och *elektricitet* men E.ON:s viktigaste aktivitet, baserat på deras mål, är *verksamhet* och den är inte beroende av *fjärrvärme* och *elektricitet*.

8 Diskussion

I detta kapitel diskuteras för- och nackdelar med beräkningsmetoden och datorverktyget samt antaganden kopplade till dessa. Kapitlet inleds med några generella lärdomar från projektet och diskussion kring hur datorverktyget har validerats och verifierats. Därefter följer en diskussion kring befintlig data, beräkningsmetoden och datorverktyget och kapitlet avslutas med en diskussion kring flödesbegreppet.

8.1 Generella lärdomar från projektet

Under arbetets gång har ett antal olika problem uppstått och resulterat i ett antal tips för fortsatt arbete, till exempel examensarbeten.

- För att förstå behovet av litteraturstudien hade det varit önskvärt att börja med beräkningar parallellt med litteraturstudien.
- Att börja programmera MATLAB innan beräkningar på testfall genomförts är inte att rekommendera. I efterhand kan det konstateras att många onödigt komplicerade funktioner i MATLAB hade kunnat undvikas på så sätt.
- Innan programmeringen av figurer i MATLAB hade det varit lämpligt att öka detaljrikedomen i kravspecifikationen och rita samtliga figurer för handi en fallstudieför undvika att onödiga delar skapas i MATLAB. Med en tidig fallstudie och bättre kravspecifikation hade det varit enklare att upptäcka praktiskt önskvärda funktioner i datorverktyget.
- Att dela upp arbetet i etapper har varit bra för att tidigt tydliggöra vad som måste göras men det har inte följts under projektets gång, då den visade sig vara opraktisk att följa. Däremot fungerade den detaljerade tidsplanen som upprättades i samband med etapperna bra, framförallt för att följa upp vad som behöver göras.
- Tidigt i arbetet med litteraturstudien upplevdes supply chain risk management som ett område med tankesätt och metoder liknande de som behövs för flödesberoenden och därmed något som skulle kunna ligga till grund för detta arbete. Tidigt visade det sig dock vara svårt att hitta information om kvantitativa metoder. De kvantitativa metoderna är ofta delar av kommersiella verktyg och därmed finns inte beräkningsmetoder fritt tillgängligt.

8.2 Diskussion kring validering och verifiering

Då metoder för validering beskrevs i kapitel 3.3 var målsättningen att presentera metoder för validering som kan vara relevanta för datorverktyget. Samtidigt var det tidigt tydligt att större delen av dessa valideringsmetoder tar mer tid än vad som är rimligt för att vara en del av examensarbetet. En metod som varit användbar men då framförallt för verifiering är att köra testfall där det går att beräkna resultaten med manuella beräkningar vilket presenterats i kapitel 6.3. I kapitel 3.3 beskrevs att användningen av ett par olika testfall kan vara lämpligt, men här har ett testfall använts som anpassats för att vara så heltäckande som möjligt. Att jämföra ett testsystem med ett verkligt system undersöktes kort men användes inte på grund av att det var ett för stort antal olika faktorer som påverkade resultatet i det i verkliga systemet som inte gick att förklara på ett överskådligt sätt med testsystemet. Därmed gick det inte att på ett enkelt sätt jämföra systemen och inte heller att förklara skillnaderna. Att jämföra resultat från datorverktyget med expertbedömningar övervägdes men vi kom fram till att experterna själva bör ha mycket svårt att göra korrekta bedömningar i och med att systemet består av flera indirekta beroenden, vilket också knyter an till behovet av ett analysverktyg. Detta tillsammans med att tiden som fanns tillgänglig för genomförandet av examensarbetet var begränsad, har gjort att validering i form av expertbedömningar inte har använts.

De övriga valideringsmetoderna som togs upp i kapitel 3.3 men som inte undersökts ytterligare, är jämförelse med historisk data eller med andra modeller. Detta eftersom dessa båda kräver omfattande förarbete med insamling av historisk data alternativt anpassning av andra modeller.

Det är verifierat att datorverktyget utför beräkningar så som dessa specificerats och i enighet med antaganden i kap 5.1, både genom strukturerade genomgångar av koden och utvärdering av ett omfattande testfall. Fallet är tänkt att kontrollera så många funktioner som möjligt i datorverktyget men samtidigt vara möjligt att överskåda. En stor arbetsinsats har varit att göra manuella jämförande beräkningarna för att verifiera datorverktyget, ett par förenklingsförsök har testats men förkastats då modellen visade sig vara för komplex. Det har under verifieringen varit tydligt att datorverktyget är mycket känsligt för indatafel. Till exempel har det, under utvecklingen och verifieringen, både uppstått fel som är lätta att identifiera, med felmeddelande från datorverktyget, och mer svåridentifierade fel identifierats, som att datorverktyget ger felaktiga resultat. För att undvika vissa av dessa problem skapades funktionen *Indatacheck* som kontrollerar att data som matas in i datorverktyget är korrekt formaterat och det rekommenderas att den används innan indata analyseras vidare. Funktionen *Indatacheck* är ett resultat av problem som uppstått under utvecklingen och inte en heltäckande kontroll av inmatad data.

8.3 Diskussion kring indata

I följande kapitel diskuteras möjliga brister i befintlig indata och hur flöden kan kopplas till geografisk information.

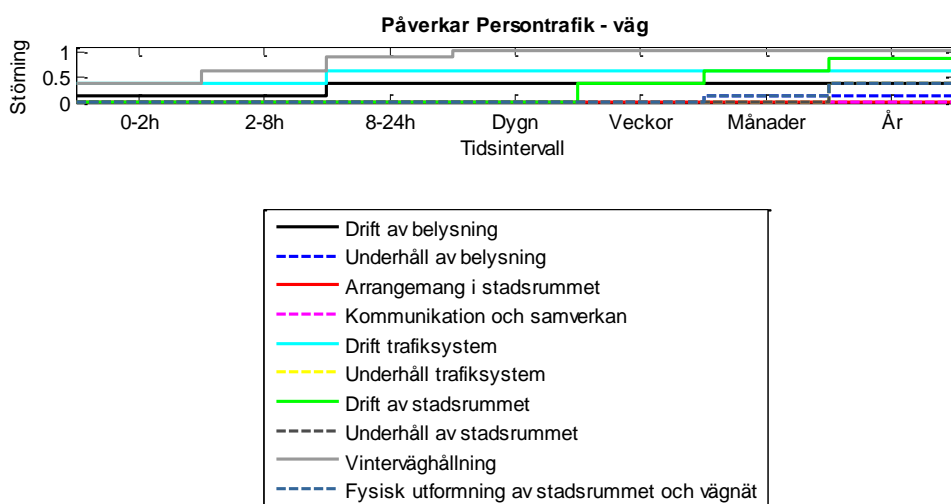
8.3.1 Brister i befintlig indata

Det finns ett par problem och brister med den data som finns tillgänglig från aktörerna. En brist som uppstått, framförallt under den tidiga datainsamlingen, var att det inte fanns några fördefinierade flöden som aktörerna kunde välja mellan. Ett tydligt exempel på detta är barn- och skolförvaltningen som bland annat bedriver aktiviteten *tillhandahålla mat* utan att vara beroende av flödet *livsmedel*. Flödet *livsmedel* tillkom under datainsamlingen för miljöförvaltningen och bör vara ett flöde som barn- och skolförvaltningen angett ett beroende av om det varit definierat vid datainsamlingen. Med fördefinierade flöden bör liknande problem minskas. Det kan dock vara så att problemet återkommer i stora system eftersom det kan vara svårt att hitta rätt flöde i en fördefinierad lista som består av ett stort antal flöden. Detta ställer därför höga krav på att de som håller workshops och intervjuer har en bra uppfattning av vilka flöden som finns och hur dessa är avgränsade.

En kategori av flöden som upplevs som problematiska i tillgänglig indata är olika former av transporter. En fråga som uppstått är hur aktörerna har angett flödesberoenden när det gäller transporter av anställda till arbetsplatsen. Visa aktörer har angett beroende av *kollektivtrafik* och andra *person och godstrafik* eller *personal till arbetsplats*. Flödet *personal till arbetsplats* är bara kopplat till Copenhagen Malmö Port (CMP) och E.ON vilket bör bero på att flödet definierats under datainsamlingen för CMP som skedde i den andra omgången av workshops under våren 2013. Aktörer som intervjuades under hösten 2012 har istället angett ett beroende av *kollektivtrafik* och *person och godstrafik* men eftersom i princip alla aktörer bör vara beroende av att de anställda kan transportera sig till jobbet saknas antagligen beroendet av *personal till arbetsplats* ofta. De persontransporter aktören är beroende av för att utföra sina aktiviteter bör däremot fortfarande anges specifikt. Vi förespråkar att alla aktörer som är beroende av personal anger beroende av *personal till arbetsplats* och det införs en ny aktör som representerar privatpersoner. Aktören privatpersoner skulle kunna baseras på en kartläggning av beroenden och påverkan, antingen på intervjuer med ett större antal personer eller genom statistik över bland annat transportvanor. Aktören privatpersoner skulle till exempel vara beroende till en viss del av *kollektivtrafik* och till en viss del av *transporter – väg*.

Att införa aktören privatpersoner är en möjlig lösning på andra problem kopplat till aktörers påverkan och beroenden. I den data som finns tillgänglig idag har ingen aktör angett beroende av till exempel *kultur – information, omsorg – unga* och *utbildning – unga*. Aktören privatpersoner borde vara beroende av flera av dessa flöden, bortfall av *omsorg – unga* och *utbildning – unga* kommer på kort sikt tvinga föräldrar att stanna hemma från arbetet och på lång sikt påverka genom att utbildningsnivån i samhället sjunker. Olika flöden kopplade till kultur kan också påverka privatpersoner genom att till exempel minska deras välbefinnande. Att flöden på olika sätt kan tydliggöra en koppling mellan hårda aspekter som tillgången på vatten eller transporter och mjuka aspekter som utbildning eller privatpersoners välbefinnande upplever vi som en stor fördel med begreppet flöden. Samtidigt kan det vara svårt att styra i vilken omfattning data om mjuka aspekter bör samlas in och att avgöra om det till exempel påverkar privatpersonens förmåga att utföra sina arbetsuppgifter.

Påverkan på flöden som bara är kopplade till en årstid eller på annat sätt bara kan inträffa under en viss tid är ett annat område som innebär vissa svårigheter. I tillgänglig indata syns det tydligaste exemplet på detta i figur 43 där aktören gatukontoret angett 100 % påverkan på flödet *persontransport - väg* om aktiviteten vinterväghållning slås ut under längre tidsperioder än ett dygn. Att vinterväghållning bara är aktuellt vintertid framgår tydligt hos Johansson och Åhsberger (2012). Samtidigt finns det ingen aspekt av metodiken eller datorverktyget som kan ta hänsyn till specifika tider för påverkan, vilket gör att samma flödespåverkan används för hela året. Detta leder till missvisande resultat då gatukontoret hela året har en markant högre påverkan på flödet *transporter - väg* än vad de övriga aktiviteterna resulterar i. Det bör skapas en enhetlig bild av hur stor del av året som påverkan från en aktivitet ska inträffa för att räknas med. Vår uppfattning är att aktiviteter som utförs ungefär en fjärdedel av året som vinterväghållning kan vara relevanta men därefter anser vi att aktiviteten utförs för sällan. En utveckling av detta kan vara att aktörerna får ange vilka årstider aktiviteterna utförs. Detta bör dock utvärderas vidare eftersom det skulle öka tidsåtgången för datainsamling ytterligare.



Figur 43. Gatukontorets påverkan på flödet persontransport - väg. Figuren en uppförstoring av figur 35 i fallstudien.

När data från flertalet viktiga aktörer i en kommun är insamlad kommer det vara viktigt att på ett enkelt sätt ta reda på när insamlingen genomfördes. Det kommer vara intressant att veta eftersom den data som är insamlad kommer behöva uppdateras med jämna mellanrum. Därför föreslår vi att insamlingsdatumet ska vara en del av informationen som finns kopplad till varje aktör.

8.3.2 Flöden och geografisk information

Att kartlägga flödens påverkan i flera olika geografiska områden, till exempel mellan olika kommuner, är problematiskt. Som flöden definierats av H. Johansson och Åhsberger (2012) kan det antingen tolkas som

om ett flöde gäller för hela regionen eller bara det som omfattas av kommunens geografiska områdesansvar. Om ett flöde anses gälla för kommunens geografiska område kan det uppstå orimliga resultat när insamlad data aggregeras upp på regional nivå. Till exempel kan det vara orealistiskt att gatukontoret i en kommun i södra änden av en region starkt påverkar flöden i norra änden av regionen. Samtidigt är det få aktörer som enbart är kopplade till aktörer inom den egna kommunen. Till exempel bör aktören Sveriges åkeriföretag påverka flödet *godstransporter* i fler geografiska områden än enbart en kommun. Datorverktyget har enbart använts för att analysera en kommun i taget och om indata för flera kommuner använts samtidigt hade flödena behandlats som gemensamma för kommunerna. Om all tillgänglig indata används hade till exempel aktören CMP (Copenhagen Malmö Port) starkt påverkat *godstransporter* i hela det undersökta området även om hamnen antagligen inte hanterar alla *godstransporter* i regionen.

För att hantera problematiken kring flödenas geografiska koppling föreslås att samma flödesindelning används för alla kommuner men med unika geografiska flöden för varje kommun. För att hantera att en aktör kan påverka andra områden än sin egen kommun kan aktören vid datainsamlingen ange för vilka geografiska områden som den har en påverkan på flöden. Förslagsvis sker denna uppdelning genom att aktören anger om påverkan gäller kommunen, andra kommuner, hela regionen, nationellt eller till och med internationellt. Om en aktör anger stark påverkan på flödet *elektricitet* i både Malmö och Lunds kommun innebär det alltså att aktören automatiskt påverkar de två separata geografiska flödena *elektricitet* (Lund) och *elektricitet* (Malmö). Genom att låta aktörer välja geografisk koppling på sin påverkan och dess påverkan därefter appliceras på flöden i alla angivna geografiska områden behöver inte geografiska kopplingar anges för beroenden. Förslaget innebär till exempel att en analys av Lunds kommun kräver ytterligare information från aktörerna i Malmö kommun. Den ytterligare data som hämtas in visar vilka aktörer i Malmö kommun som påverkar flöden i Lunds kommun. På detta sätt går det att hantera interkommunala störningar utan att behöva ha specifika flöden som kopplar samman alla kommunerna.

Vid datainsamling under hösten 2013 utförd av Lindström och Wikman (Under tryckning) har flödespåverkans geografiska koppling tydliggjorts. Resultatet från denna datainsamling kommer att presenteras under senhösten 2013 och har delvis använts i detta examensarbete. Datorverktyget har dock inte utvecklats för att specifikt ta hänsyn till geografiska kopplingar och därför har inte den delen av Lindström och Wikmans (Under tryckning) resultat använts eller utvärderats.

8.4 Utvärdering av beräkningsmetod

Den beräkningsmetod som används för att analysera beroenden och påverkan anses generellt sett vara lämplig även om den bygger på flertalet antaganden, vilka är beskrivna i kapitel 5.1. Den valda beräkningsmetoden, multiplikationsmetoden, har brister men med tillgänglig data har metoden ansetts vara lämpligast. Då det till exempel aldrig är kartlagt vad störningar i flöden praktiskt innebär för flödet betyder det att vissa av gjorda antagande bör vara mindre lämpliga. Ett exempel på detta är när en aktör är starkt beroende av flödena *elektricitet* och *dricksvatten* och en störning på båda flödena uppstår. Då är det oklart vad som händer med aktören. Med bristande tillgång på både el och vatten är det möjligt att en aktör blir helt utslagen i verkligheten men i beräkningar med datorverktyget kommer den enbart att ha sänkt kapacitet i enlighet med multiplikationsmetoden.

För att få en mer heltäckande bild över hur aktörer och flödens kapaciteter förändras med olika grader av störning skulle en omfattande datainsamling behöva genomföras. Varje aktör skulle behöva beskriva hur sin påverkan och sitt beroende är för olika grader av utslagningar. Det skulle krävas betydligt mer tid för att genomföra en sådan workshop och det skulle troligen vara svårare för aktörer att ta ställning till dess påverkan och beroende. Redan de workshops som genomförts av H. Johansson och Åhsberger (2012) tog fyra timmar och med ytterligare tidsåtgång kan det vara svårt att motivera aktörerna att delta.

Datorverktyget hade dessutom behövt modifieras för att använda en beräkningsmetod som är bättre lämpad än den linjära modellen som nu används.

Den koppling mellan aktörers mål och dess aktiviteter som används för att beräkna aktörens totala kapacitet bör skilja sig åt för olika aktörer. Detta leder till att antagandet om hur aktörens kapacitet beror av aktiviteternas kapacitet stämmer olika väl för olika aktörer. Om till exempel Sveriges Åkeriföretag inte kan utföra aktiviteten *utföra transporter* blir deras kapacitet externt fullständigt utslaget och det spelar ingen roll att de kan genomföra aktiviteten *schemaläggning*. Aktören barn- och skolförvaltningen är ett annat exempel då de kommer ha relativt hög kapacitet kvar trots att aktiviteten *bedriva undervisning* är utslaget. Hur hög kapacitet barn- och skolförvaltningen kommer få vid utslagningen varierar i stor utsträckning på antalet aktiviteter som de har angivet. Desto fler aktiviteter en aktör har angivet desto mindre påverkan uppstår vid bortfall av enskilda aktiviteter. Det går att samla in data som ytterligare kopplar ihop aktörernas mål och aktiviteter men detta bör vara ett omfattande arbete och antagligen inte praktiskt genomförbart.

Antagandet om fullständig linjäritet är nödvändigt för att datorverktyget ska fungera med den utvecklade beräkningsmetoden. Troligtvis är det ett antagande som inte ger en helt korrekt bild av verkligheten. Aktörerna har enbart svarat vad som händer vid en fullständig utslagning av aktiviteter och av flöden men datorverktyget slår ut enstaka procent av aktiviteter och flöden. Ett exempel är en 50 % utslagning av flödet *personal till arbetsplats*. Då kan det finnas en möjlighet för aktören att prioritera mellan vilka aktiviteter som behöver genomföras. Med en prioritering av aktörens aktiviteter skulle det till exempel vara möjligt för aktören att under en kortare tid upprätthålla en aktivitet helt samtidigt som en annan slås ut helt även om de angetts ha samma flödesberoenden. Hade det funnits information kring aktörens prioriteringar, likt de som finns angivna för aktörens mål, hade en implementering av detta varit relativt enkel i beräkningsmetoden och datorverktyget. Ett annat exempel på detta olinjära samband skulle även kunna gälla för en 50 % utslagning av flödet *elektricitet*. I verkligheten skulle det kunna betyda att elektriciteten är 100 % utslagen halva dygnet eller att flödet för en enskild aktör bara kan vara fullt fungerande eller helt utslaget.

Att beroenden och påverkan angetts i intervall gör att det går att presentera en osäkerhetsspridning för resultatet. Som visas i figur 25 och figur 39 finns det en spridning i resultatet då lägsta och högsta värden i det bedömda intervallet för påverkan och beroenden använts. Då figur 25 visar på en minskande spridning med ökande utslagning och figur 39 visar en oförändrad spridning även med ökande utslagning går det inte att dra någon generell slutsats av spridningsintervallens betydelse. Det kan anses att det hade varit lämpligare att göra konservativa beräkningar och använda de högsta värdena för beroende och påverkan. Om beräkningsmetoden används för att jämföra olika störningar och åtgärder bör valet av värden inte vara avgörande eftersom valet av värde bör påverka alla undersökta scenarier på samma sätt. Används däremot beräkningsmetoden för att analysera ett specifikt scenario, till exempel för att avgöra hur kritisk en viss aktör är, gör valet av värde skillnad. Vi har valt att använda medelvärdet eftersom aktörerna antagligen inte avsett något specifikt del av intervallet. När medelvärdet används anser vi att resultatet representerar medelkonsekvenserna som skulle uppstå. Det är dock viktigt att undersöka och ta hänsyn till spridningen i resultaten och *worst-case* scenarier.

I huvudsak skapar datorverktyget figurer som presenterar beroendegraden med en procents noggrannhet. Då aktören angett beroende och påverkan i intervall med en bredd på upp till 25 % är resultaten inte så exakta som en procent, vilket värdena i figurerna antyder. Att istället införa en femgradig skala har övervägts eftersom det skulle representera spridning i indata tydligare. Ett problem med att avrunda exaktheten i datorverktygets resultat till en femgradig skala är att mycket information försvinner. Till exempel kan två beroenden på 29 % respektive 31 % få värde 2 respektive 3, i en femgradig skala, och uppfattas som olika kritiska även om de egentligen är mycket snarlika.

En begränsning med beräkningsmetoden är att den är känslig för avsaknad av viktiga aktörer. Saknas en viktig aktör kan det leda till att ett stort antal beroenden inte upptäcks och att andra aktörer antingen blir för viktiga eller för oviktiga i analysresultatet. Det kan vara svårt att avgöra när ett tillräckligt antal aktörer identifierats, men en indikation kan vara att de flöden som visas i matrisen över flödets beroende av andra flöden, figur 21, saknar beroenden. Om ett flöde saknar påverkan på övriga flöden och dessutom inte blir påverkat av något annat flöde i matrisen krävs antagligen ytterligare datainsamling. Lindström och Wikman (Under tryckning) har utvecklat metodik för att bedöma om antalet aktörer är tillräckligt och rekommenderas för utförligare diskussioner kring problematiken.

8.5 Utvärdering av datorverktyg

Under utveckling av datorverktyget har flera fördelar med datorverktyget identifierats. Datorverktyget illustrerar på ett tydligt vis kopplingar mellan aktörer som inte har direkta beroende mellan sig. Detta har inte tidigare varit möjligt och det möjliggör för aktörer att identifiera och införa stötdämpare för beroenden som tidigare inte varit kända. Till exempel kan aktörer med flertalet kopplingar till andra aktörer identifieras då dessa kan vara viktiga i samhället eller extra känsliga för flera olika störningar. Dessa aktörer kan därför vara viktiga att studera vidare. Datorverktyget har funktioner som på ett tydligt sätt visar styrkan på beroendet mellan olika aktörer och kan vara till hjälp i arbetet med att ta fram de mest kritiska aktörerna.

I kapitel 0 presenteras de figurer som datorverktyget kan skapa. Under respektive figur finns en beskrivning av vad som visas i figuren samt en avstämning mot de krav som är specificerade i kravspecifikationen, se appendix a. Med ett undantag anses samtliga krav som är specificerade vara uppfyllda. Undantaget är en figur som visar hur aktörernas mål påverkas av flödesstörningar. Figuren finns inte med i datorverktyget dels på grund av tidsbrist och dels på grund av att det är oklart hur målen skulle presenteras. Under fallstudien har det också framkommit ett behov av en figur som visar aktörers påverkan på flöden men denna funktion har utelutits på grund av tidsbegränsningar.

8.6 Generell utvärdering av flödesbegreppet

Att använda flöden för att identifiera beroenden upplevs ha ett antal tydliga fördelar. Genom att datainsamlingen genomförts med expertbedömningar upplever vi att det går att identifiera aktörers kopplingar till flöden som annars hade varit svåra att identifiera. Alternativet hade varit att försöka anpassa kvantitativ data från andra källor, till exempel ekonomisk data.

Vi anser att med de modeller och metoder för beroendeanalys som är beskrivna i kapitel 2.6 är det mer komplicerat att beskriva flöden som saknar koppling till fysisk infrastruktur, till exempel utbildning, vilket möjliggörs med flödesbegreppet. Att aktörer redan i samband med insamlingen av data blir mer medvetna om sina direkta beroenden och sin påverkan i samhället ses också som en fördel. Flödesbegreppet tillåter att beroenden och påverkan som uppstår på längre tidsperioder enkelt kan beskrivas. Utan tidsaspekten hade beroenden mellan aktörer blivit missvisande, en aktör som är beroende av ett flöde på kort sikt hade blivit beroende av en aktör som påverkar ett flöde på lång sikt även om detta inte behöver vara fallet. Till exempel genom att aktören efter en viss tid kan byta leverantör eller upprätthålla aktiviteten på annat sätt.

Det har även identifierats några nackdelar med flödesbegreppet under arbetets gång. En av dessa är att resultatet helt bygger på expertbedömningar. Det hade varit lämpligt att antingen verifiera dessa bedömningar med andra experter eller med statistiska uppgifter. De workshops som utförs har av Johansson och Åhsberger har syftat till att samla alla aktörens tillgängliga experter (J., Johansson, personlig kommunikation, 25 november, 2013). Därför kan det vara svårt att hitta experter som är tillräckligt insatta för att verifiera uppgifterna från workshopsen. I framtiden kan det vara lämpligt att i enstaka fall dela upp experterna i två grupper för att kunna jämföra deras resultat. Ett annat sätt att hantera osäkerheten i

expertbedömningarna kan vara att experterna anger hur säkra de är på sina bedömningar och detta kan vägas in i resultatet enligt metodiken som beskrivs av Setola och De Porcellinis (2008).

Generellt sätt upplevs tidsperioder som en styrka med flödesbegreppet. Att datainsamlingen tar hänsyn till olika tidsperioder har även brister eftersom det kan vara svårt för aktörer att avgöra under vilken tidsperiod som påverkan på flöden kommer ske. Att det kan vara svårt att ange tidsperiod korrekt kan påverka resultatet, vilket är något som bör tas i åtanke. Andra nackdelar med flödesbegreppet är att det är ett nytt begrepp inom risk- och sårbarhetsanalys. Det gör att det kan vara svårt för aktörer att förstå och tillämpa begreppet.

Ett intressant tillägg hade varit att veta vilka mål som kommunen har på ett övergripande plan. Med den typen av information skulle olika aktörer kunna prioriteras och analysen blivit en bättre representation av verkligheten. Det hade till exempel varit möjligt att utveckla datorverktyget för att presentera hur kommunens mål påverkas vid en utslagning av ett flöde eller en aktör. Detta tillägg är också något som är en del av den metodik som finns utvecklad. Däremot har den hittills inte använts vid datainsamling (J., Johansson, personlig kommunikation, 25 november, 2013).

Flödesbegreppet har sin bakgrund i ett samarbete mellan LUCRAM och Resilient Regions Association. Inom ramen för detta samarbete finns målsättningen att det ska utföras en fallstudie som visar på användbarheten i flödesbegreppet. Fallstudien som presenterades i kapitel 7 är en ansats till detta. Dock saknas fortfarande mycket indata och formatet för indata behöver utvecklas i enlighet med tidigare diskussioner. Det avråds därmed från att använda resultaten från fallstudien vidare, t.ex. för att identifiera vilka aktörer som faktiskt är mest kritiska i kommunen. Däremot visar fallstudien tydligt både begränsningar och exempel på slutsatser som skulle kunna göras med datorverktyget om väl indata kommer på plats.

9 Slutsats

Projektets övergripande mål har varit att skapa en beräkningsmetod för att aggregera och analysera flödesdata på en högre nivå än enskild aktör. Detta mål anses vara uppfyllt då datorverktyget som utvecklats kunnat användas för att utföra en förenklad fallstudie för Lunds kommun. Aggregeringen utförs med en egenutvecklad beräkningsmetod där sammanslagning av påverkan utförs med multiplikation. Aktörernas kapacitet efter påverkan baseras på de enskilda aktiviteternas kapacitet och påverkan på aktörens mål.

Projektets mål från kapitel 1.3 presenteras nedan och därefter en bedömning av måluppfyllnaden.

Beskriva befintliga modeller som kan ligga till grund för utvecklandet av en lämplig metod som svarar mot övriga mål inom detta examensarbete, ett exempel på en sådan modell är input-output modellen.

Befintliga modeller har undersökts och presenterats i kapitel 2.6. Ingen av modellerna har uppfyllt de krav som tagits fram för beroendeanalys med hjälp av flöden. Därför har en egenutvecklad beräkningsmetod använts.

Bestäm om den utvecklade beräkningsmetoden är lämplig för att hantera aggregering av flöden.

Den utvecklade beräkningsmetoden fungerar väl för aggregering av flöden om de antaganden som gjorts är korrekta. För att säkerställa att antagandena är korrekta måste flödesbegreppet utvecklas vidare. De två huvudsakliga bristerna med flödesbegreppet anses vara att hur linjäriteten i en störning ska hanteras samt oklara flödesdefinitioner.

Implementera den utvecklade metoden i ett datorverktyg för att genomföra analyser.

Den utvecklade beräkningsmetoden har omsatts i ett datorverktyg som kan användas för att genomföra analyser. Datorverktyget är däremot inte direkt användbart för slutanvändare eftersom det saknas användargränssnitt för analys och datainmatning.

Att undersöka och identifiera de tilltänkta användarnas behov kopplat till datorverktyget.

Med hjälp av tilltänkta användare i Lunds kommun har användarbehov identifierats och omsatts till en kravspecifikation.

Utvärdera det framtagna datorverktyget utifrån identifierat behov.

Datorverktyget uppfyller den kravspecifikation som skapats tillsammans med tilltänkta användare förutom för en funktion. Den funktion som finns i kravspecifikationen men som inte skapas i datorverktyget är en figur över flödets påverkan på aktörernas mål.

9.1 Förslag på fortsatt arbete

De förslag på fortsatt arbete som tagits fram delas upp i fortsatt utveckling av datorverktyget och metodiken kring beroendeanalys med hjälp av flöden. Samtliga av dessa förslag presenteras nedan.

9.1.1 Fortsatt utveckling av datorverktyget

- För att möjliggöra analys av ett större antal aktörer och mer verklighetsbaserade störningar, där till exempel en hel samhällssektor störs samtidigt, föreslår vi att alla aktörer ska kategoriseras i sektorer. Exempel på kategorier som kan användas är transporter och sjukvård.
- En studie som undersöker linjäritetsantagandet i kapitel 5.1.3 bör genomföras. Den bör dessutom testa vilken av additions- eller multiplikationsmetoden i kapitel 5.2 som bäst stämmer överens med verkligheten.
- För att underlätta analys av resultatet från datorverktyget bör flera av funktionerna förändras så att kopplingen mellan flöde, aktör och enskild aktivitet i framgår direkt. Detta diskuterades i kapitel 7.4.2.
- För att underlätta analys av datorverktygets resultat bör någon av funktionerna kunna visualisera hur aktörernas aktiviteter kopplar till målen. Se beskrivning i kapitel 8.4.
- Det finns ett intresse hos Resilient Regions Association och företaget iFACTS att koppla ihop det utvecklade datorverktyget med ett existerande ledningssystem vilket kan öka datorverktygets möjligheter att användas. Kopplingen diskuterades under examensarbetet men konkreta försöks gjordes inte då skillnaderna mellan datorverktyg och ledningssystem bedömdes vara för stora.
- De geografiska kopplingar som beskrevs i kapitel 8.3.2 bör implementeras i datorverktyget. När verktyget kan ta hänsyn till dessa kan ett lämpligt steg vara att analysera hur aggregering på till exempel länsnivå kan utföras. Det krävs omfattande data för att möjliggöra aggregering på länsnivå och det finns ett antal andra delar av flödesbegreppet som måste utvärderas ytterligare innan det är lämpligt att påbörja detta arbete.
- I datorverktyget är det svårt att dra slutsatser som gäller över flera tidsperioder eller hur beroende och påverkan förändras över tiden. Det hade till exempel varit önskvärt med en animation för spridningskedjor eller att se hur matrisen för aktörers beroende av aktörer förändras med tiden.

9.1.2 Förslag till fortsatt arbete kring beroendeanalys genom flöden

Några förslag på fortsatt arbete kopplat direkt till begreppet flöde har även framkommit.

- Jämföra analys av flödesberoenden med verkliga händelser eller andra modeller som till exempel input-output-modeller eller agentbaserade modeller.
- Utveckla begreppets koppling till olika tidsberoende händelser. I kapitel 7.4.2 beskrivs säsongbetonade aktiviteter som vinterväghållning men även hur till exempel räddningstjänstens aktiviteter som *Skadeavhjälpande insats* hanteras är intressant.
- För att komma till rätta med att många flöden, som i verkligheten är viktiga för samhället, saknar beroende aktörer föreslår vi att aktören privatpersoner införs. Förutom att aktören privatpersoner visar att fler flöden är viktiga skulle aktören även lösa andra problem som till exempel hur personal kommer till arbetsplatsen. Detta eftersom aktören privatpersoner skulle vara beroende av både *kollektivtrafik* och *transport – väg*. Aktören privatpersoner kan skapas på olika vis, till exempel på tidigare insamlad statistisk eller på intervjuer med personer. Aktören privatpersoner diskuteras i kapitel 8.3.
- Utveckla metodiken kring flöden för att ta hänsyn till samhällets övergripande mål, vilket beskrivs kort i kapitel 6.2.5.

10 Litteraturförteckning

- Abrahamsson, M. (2009). *Analytic input to societal emergency management-on the design of methods*. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- Ahlin, L., Ardefel, A., & Sundström, L. (2005). *Alla blir vinnare*. Göteborg: Göteborgs universitet.
- Department of Defense. (den 08 08 2007). *Department of Defense Directive*. Hämtat från DoD Modeling and Simulation (M&S) Management: <http://www.dtic.mil/whs/directives/corres/pdf/500059p.pdf> den 10 10 2013
- Dudenhoefter, D. D., Permann, M. R., & Manic, M. (2006). CIMS: A framework for infrastructure interdependency modeling and analysis. *Proceedings of the 38th conference on Winter simulation*. Washington, DC: Winter Simulation Conference.
- Ginstrup, I., & Rubil, M. (2013). *Hur godsflödet genom Göteborgs Hamn påverkar sektorer i Sverige*. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- Haines, Y., & Jiang, P. (2001). Leontief-Based Model of Risk in Complex Interconnected Infrastructures. *Journal of Infrastructure Systems*, 7(1), 1-12.
- Hassel, H. (2010). *Risk and vulnerability analysis in society's proactive emergency management : developing methods and improving practices*. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- Höst, M., Regnell, B., & Runeson, P. (2006). *Att genomföra examensarbete*. Lund: Studentlitteratur.
- Idaho National Laboratory. (u.d.). *Modelling and simulation*. Hämtat från Idaho National Laboratory: https://inlportal.inl.gov/portal/server.pt/community/national_and_homeland_security/273/modeling_and_simulation/1707 den 2013 10 03
- ISO. (2009). Svensk standard SS-ISO 31000:2009.
- Johansson, H., & Åhsberger, A. (2012). *Beroendeanalys ur ett flödesperspektiv - jämförelse av metoder för datainsamling*. Department of Fire Safety Engineering and Systems Safety. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- Johansson, H., Jönsson, H., & Johansson, J. (2007). *Analys av sårbarhet med hjälp av nätverksmodeller*. Lund: Lund University Centre for Risk Assessment and Management .
- Johansson, J. (2010). *Risk and Vulnerability Analysis of Interdependent Technical Infrastructures: Addressing Socio-Technical Systems*. Lund: Department of Measurement Technology and Industrial Electrical Engineering, Lund University.
- Johansson, J. H. (2011). Vulnerability analysis of interdependent critical infrastructures: case study of the Swedish railway system. *International Journal of Critical Infrastructures*, 7(4), 289-316.
- Johansson, J., Svegrup, L., & Hassel, H. (2013). Societal consequences of critical infrastructure vulnerabilities; integrating power system and regional inoperability input-output models. *ESREL2013*. Amsterdam: ESREL2013.
- KBM. (den 31 08 2005). *Krishantering i stormens spår; sammanställning*. Hämtat från <http://ndb.msb.se/Document/Report/633275274554050000.pdf> den 02 09 2013

- Kosterev, D. N., Taylor, C. W., & Mittelstadt, W. (1999). Model validation for the August 10, 1996 WSCC system outage. *Power Systems, IEEE Transactions on*, 967-979.
- Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis*. (2. ed.). New York: McGraw-Hill.
- Lindström, A., & Wikman, M. (Under tryckning). *Kartläggning och aggregering*. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- Little, R. G. (2002). Toward More Robust Infrastructure: Observations on Improving the Resilience and Reliability of Critical Systems. *Hawaii International Conference on System Sciences (HICSS'03)*.
- Lunds kommun. (2012). *Risk och Sårbarhetsanalys för Lunds kommun*. Lund: Lunds kommun.
- Macal, C. M., & North, M. J. (den 29 11 2006). *Introduction to Agent-based Modeling and Simulation*. Hämtat från <http://www.mcs.anl.gov/~leyffer/listn/slides-06/MacalNorth.pdf> den 04 09 2013
- MSB. (2007). *Beroendeanalys - så gör du!* Karlstad: Myndigheten för samhällsskydd och beredskap (MSB).
- MSB. (2009). *Faller en - faller då alla? : en slutredovisning från KBM:s arbete med samhällskritiska beroenden : en sammanfattning av rapporten*. Karlstad: Myndigheten för samhällsskydd och beredskap (MSB).
- MSB. (den 30 06 2011). *Beroendehjulet*. Hämtat från Myndigheten för samhällsskydd och beredskap (MSB): <https://www.beroendehjulet.se/> den 03 10 2013
- MSB. (2011a). *Ett fungerande samhälle i en föränderlig värld : nationell strategi för skydd av samhällsviktig verksamhet*. Karlstad : Myndigheten för samhällsskydd och beredskap (MSB), 2011.
- MSB. (2011b). *Vägledning för Risk- och sårbarhetsanalyser*. Karlstad: Myndigheten för samhällsskydd och beredskap (MSB).
- North, M. J. (2001). Multi-agent social and organizational modeling of electric power and natural gas markets. *Computational & Mathematical Organization Theory*, 331-337.
- Parunak, H. V., Savit, R., & Riolo, R. L. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. *Multi-Agent Systems and Agent-Based Simulation*, 10-25.
- Paulsson, U. (2007). *On managing disruption risks in the supply chain : the DRISC model / by Ulf Paulsson*. Lund : Department of Industrial Management and Logistics, Engineering Logistics, Lund University, 2007 (Lund : KFS).
- Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Critical Infrastructure Interdependencies. *IEEE Control Systems Magazine*, 11-25.
- Sapkota, P. (2010). *A Thesis entitled Modeling Diffusion Using an Agent-Based Approach*. Toledo: The University of Toledo.
- Sargent, R. G. (2005). Verification and validation of simulation models. *WSC '05 Proceedings of the 37th conference on Winter simulation* (ss. 130-143). Washington, DC: Winter Simulation Conference.
- SCB. (den 14 08 2013). *Befolkningsstatistik*. Hämtat från Statistiska Centralbyrån: http://www.scb.se/Pages/TableAndChart____228189.aspx den 11 11 2013
- Setola, R. (2007). Availability of healthcare services in a network-based scenario. *International journal of networking and virtual organisations*, 4(2), 130-144.

- Setola, R., & De Porcellinis, S. (2008). A methodology to estimate input-output inoperability model parameters. i *Critical Information Infrastructures Security* (ss. 149-160). Springer.
- Statens energimyndighet. (2005). *Stormen Gudrun – Konsekvenser för nätbolag och samhälle*. Eskilstuna: Statens energimyndighets förlag.
- Swedish Standards Institute. (2009). *Risikhantering – Principer och riktlinjer (ISO 31000:2009, IDT)*. Stockholm: Swedish Standards Institute.
- Svegrupp, L. (2012). *The Inoperability I/O model*. Lund: Department of Fire Safety Engineering and Systems Safety, Lund University.
- System Dynamics Society. (den 17 02 2011). *The Field of System Dynamics*. Hämtat från System Dynamics Society: http://www.systemdynamics.org/what_is_system_dynamics.html den 02 10 2013
- Zhang, P., & Peeta, S. (2011). A generalized modeling framework to analyze interdependencies among infrastructure systems. *Transportation Research Part B: Methodological*, 553-579.

Appendix A. Kravspecifikation

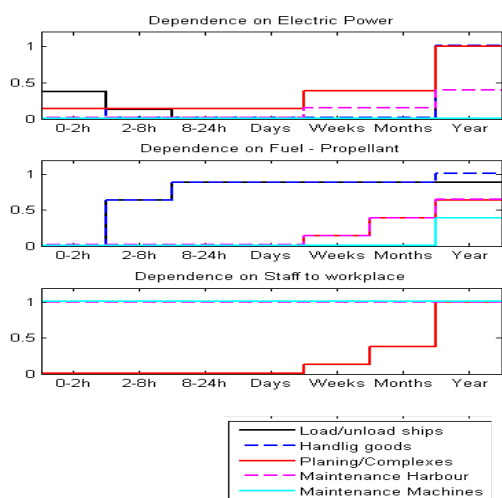
Kravspecifikationen baseras på åsikter som framkommit vid ett möte med Eva Greko och Lennart Larsson, representanter från Lunds kommun, den 4/10 – 13. Figuren som visas nedan är de figurer som presenterades i samband med mötet och åsikterna som framkom under mötet har sedan tolkats till de krav som återfinns för respektive figur. Ett övergripande krav är att alla delar av datorverkyget ska hantera system med en storlek från två flöden och två aktörer upp till obegränsat antal flöden och aktörer. Vid samtliga beräkningar, utom då indata presenteras, ska effekten av indirekta beroenden kunna analyseras. Hur väl figurerna som är skapade i datorverkyget stämmer överens med kravspecifikationen beskrivs i kapitel 0.

För att visualisera indata kring flödesberoenden och påverkan ska ett antal olika exempel på grafer och tabeller skapas. Exempelfiguren nedan uppfyller inte de krav som specificerats och presenteras enbart för att visa vad kravspecifikationen baserats på.

Aktörers fullständiga beroende av flöden

Figuren som önskas ska visa en delfigur för varje flöde och i delfiguren ska kapaciteten för alla påverkade aktörer presenteras på y-axeln och på x-axeln ska tiden presenteras. Se figur 1 för ett exempel med över grafer men med felaktigt innehåll.

- Delfigurerna ska visa i vilken utsträckning som respektive aktör är beroende av flödet i de olika tidsstegen.
- Det ska skrivas ut vilka flöden som ingen aktör är beroende av, om det finns några.
- Aktörernas kapacitet som presenteras på Y-axeln ska beräknas med hjälp av indirekta beroenden.



Figur 1 Exempel på visualisering av aktörens beroende, uppdelat på aktiviteter, över tiden. Visualiseringen som efterfrågas ska istället visa samtliga aktörer som är beroende av flödet. Visualiseringen i bilden är ett exempel på placering av grafer och linjer och visar inte det krav som ställs.

Matris över flödens beroende av övriga flöden

Figuren som önskas ska visa hur flöden är beroende av andra flöden. Se figur 2 för ett exempel med över grafer men med felaktigt innehåll.

- Resultatet ska visualiseras i matrisform med flöden på både x- och y-axeln.
- Rutorna i matrisen ska färgläggas efter hur starkt beroendet mellan flödena är.
- Varje ruta ska innehålla ett tal som beskriver hur starkt beroendet är.
- Resultatet ska visas för efterfrågad tidsperiod.

		Verksamhet						
Beroende av	Arbetskraft	Elförsörjning	Akutsjukvård	Vård/omsorg	IT-telekomm	Vatten/avlopp	Transporter/Drivmedel	Socialtjänst
Arbetskraft								
Elförsörjning								
Akutsjukvård								
Vård-och omsorg								
IT-telekomm								
Vatten/avlopp								
Transporter								
Socialtjänst								

Figur 2 Exempel på visualisering av flödets beroende av andra flöden. Från Lunds kommun (2012, s. 20). Visualiseringen i bilden är ett exempel på placering av flödes namn och färger som kan användas. I figuren saknas det siffror i rutorna.

Matris över aktörernas beroende av övriga aktörer

Figuren som önskas ska visa hur aktörer är beroende av andra aktörer, se figur 3 för ett exempel över en matris men med felaktigt innehåll.

- Resultatet ska visualiseras i matrisform med aktörer på både x- och y-axeln.
- Rutorna i matrisen ska färgläggas efter hur starkt beroendet mellan aktörerna är.
- Varje ruta ska innehålla ett tal som beskriver hur starkt beroendet är.
- Resultatet ska visas för efterfrågad tidsperiod.
- En kombination av figur 2 och 3 som visar en *matris över aktörers beroende av flöden* men i övrigt uppfyller samma krav som figur 3 ska skapas.

		Beroende av:							
		Hamn	EON	Swedish Transport	Miljöförvaltningen	Kultur och fritidsnämnden	Barn och skolförvaltningen	Renhållningsverket	Gatukontoret
Aktor	Hamn		5	5	1	1	1	3	3
	EON	2		3	1	2	1	3	3
	Swedish Transport	5	4		2	1	1	1	5
	Miljöförvaltningen	2	5	2		2	2	3	3
	Kultur och fritidsnämnden	1	5	2	3		2	3	1
	Barn och skolförvaltningen	1	5	4	3	3		2	2
	Renhållningsverket	2	4	2	5	1	2		3
	Gatukontoret	1	4	2	2	5	3	1	

Figur 3 Exempel på matris med aktörers beroende av andra aktörer. Visualiseringen i bilden är ett exempel på placering av aktörers namn och färger som kan användas.

Viktigaste aktörerna och flödena för en aktör och kommun

Figuren som önskas ska rangordna de fem viktigaste flödena och aktörerna för önskad aktör eller hela kommunen, se figur 4 för ett exempel med över grafer men med felaktigt innehåll.

- Det ska vara möjligt att välja vilken aktör som önskas presenteras.
- Resultatet från kommunen ska visas i samband med enskilda aktörer
- Rangordningen mellan olika aktörer ska baseras på att samtliga aktörer och flöden var för sig utsätts för samma utslagning och de aktörer som har störst inverkan ska redovisas.
- Resultatet ska visas för efterfrågad tidsperiod.
- Rangordningen ska representera hur viktig aktören eller flödet är för vald aktör eller kommun.

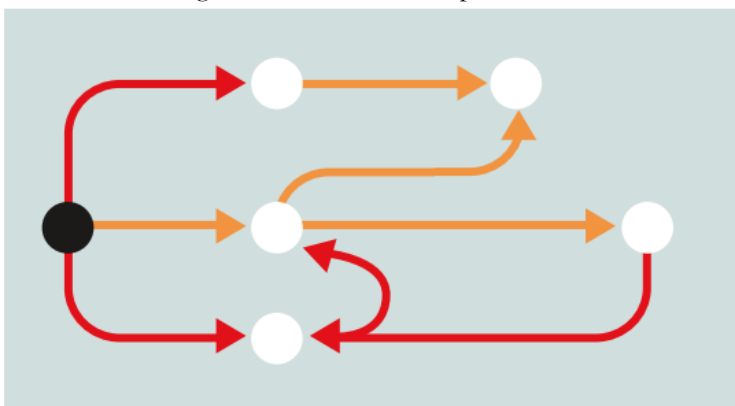
Miljöförvaltningen				Lunds kommun			
Flöde:		Aktör:		Flöde:		Aktör:	
El	5	EON	5	El	5	EON	5
Vatten	4	VA SYD	4	Vatten	5	VA SYD	5
Värme	4	Skånetrafiken	2	Värme	4	Sjukhuset	4
Kollektivtrafik	2	IT-avdelningen	2	Sjukvård	4	Räddningstjänsten	3
IT	2	Gatukontoret	1	Kollektivtrafik	2	Lunds energi	3

Figur 4 Exempel på rangordningstabell, för att visualisera de mest kritiska flödena och aktörerna. Visualiseringen i bilden är ett exempel på placering av tabeller och vad dessa skulle kunna innehålla.

Spridningskedja från en aktör till alla påverkade aktörer

Figuren som önskas ska illustrera spridningskedjor som visar vilka aktörer en utslagen aktör påverkar och vilka som de i sin tur blir påverkade av. Figuren visades inte under användarmötet men efterfrågades specifikt och återfinns därför här i figur 5.

- Samtliga steg av beroende ska illustreras och kopplingarna mellan aktörerna ska visas med pilar.
- Kapaciteten hos aktörerna ska presenteras för en vald tidsperiod efter att en aktör slagits ut.
- En alternativ bild som visar spridning- och beroendekedjor för en aktör ska skapas, utan att beräkningsmetoden och utan kapaciteter.



Figur 5 Exempel på spridningskedjor. Från MSB (2007, s. 7).

Matris över hur aktörernas mål beror av flöden

Figuren som önskas ska illustrera hur viktiga respektive flöde är för att uppfylla aktörers mål. Se figur 6 för ett exempel med över grafer men med felaktigt innehåll.

- Det ska vara möjligt att välja vilken eller vilka aktör som målen ska presenteras för.

		Beroende av															
		Electric Power	Water & Wastewater	Land & Mobile Phone	Transportation - Road	Fuel - Propellant	IT - Internal	IT - External	Public transportation	Goods - Road	Goods - To Harbour	Goods - Out Harbour	Staff to workplace	Gas	District heating	Fuel - Generation	Food
Mål	Better Environment Lund	1	4	3	3	5	4	3	3	1	5	2	1	5	5	4	5
	Good food quality	3	4	4	2	2	2	3	2	5	1	1	1	3	1	4	4
	Good drinking water	4	2	1	3	3	5	5	5	1	3	4	3	4	1	3	1
	Good air quality	5	4	2	2	2	4	1	3	5	3	4	1	1	5	1	1
	Municipal residents richer life	3	1	2	1	5	4	1	4	1	4	5	4	2	2	3	3
	Municipal residents knowledge of culture	2	5	3	4	4	4	3	3	4	4	5	3	5	3	2	3
	Culture avtivities for municipal residents	5	2	5	5	5	1	4	1	1	1	4	2	1	4	2	5
	Municipals residents possibility to effect culture	4	4	4	3	5	1	4	1	1	1	1	4	1	2	2	4
	Good School	3	2	3	3	5	3	2	2	1	3	3	2	4	4	4	2
	Good Pre-school	2	3	4	5	3	5	5	5	2	1	1	4	5	4	2	3
	Good Off-school activities	2	3	4	5	1	3	5	2	4	2	2	4	2	5	5	3

Figur 6 Exempel på matris med hur aktörernas mål är kopplade till vilket beroende de har av olika flöden. Visualiseringen i bilden är ett exempel på placering av aktörers namn och färger som kan användas.

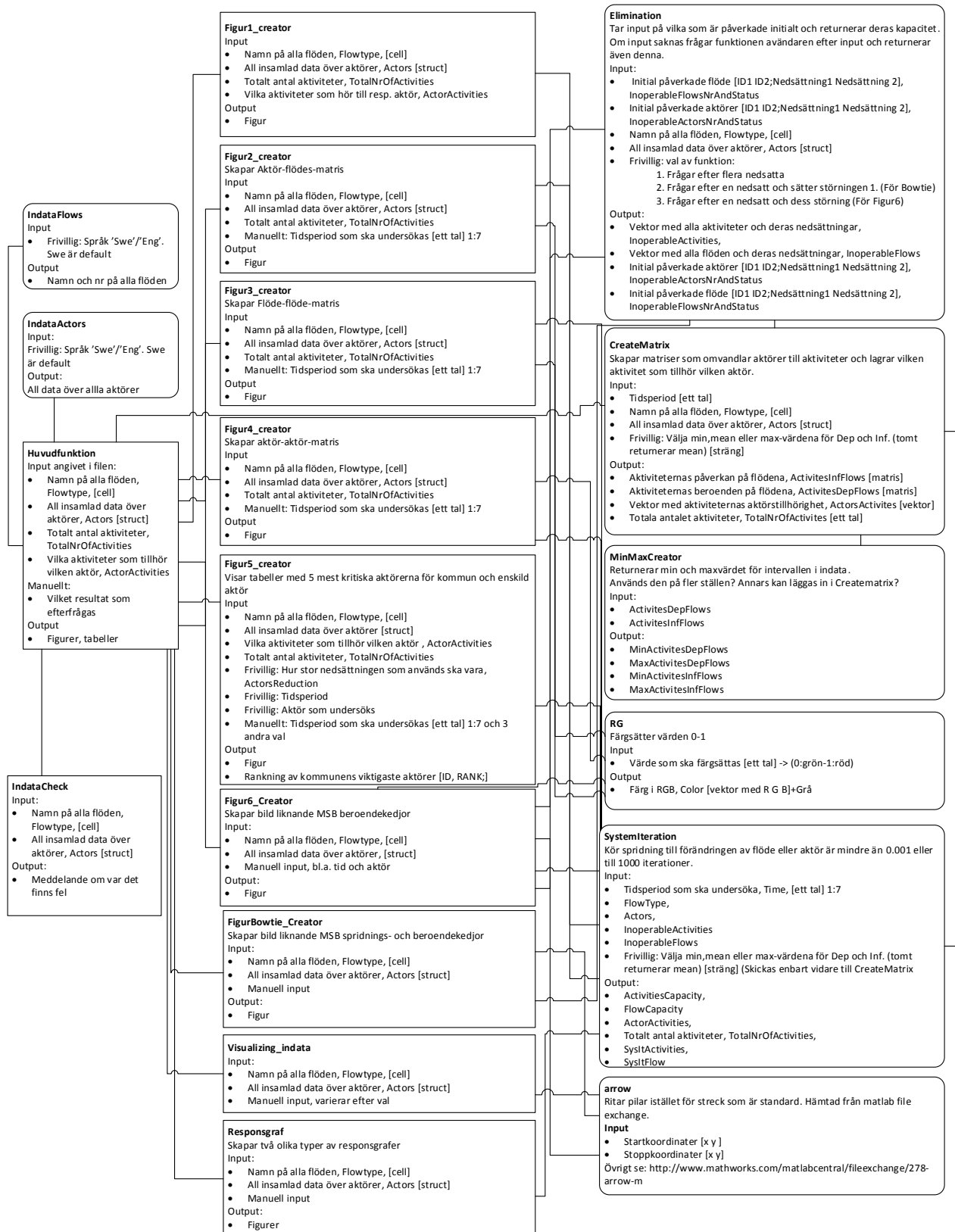
Appendix B. Funktionskarta för MATLAB-koden

Funktionskartan för koden är en beskrivning av hur funktionerna i Appendix D kallar på varandra. De funktioner som behöver eller behövs för andra funktionerna binds samman med linjer.

I funktionskartan finns det tre olika typer av input:

- Den som krävs för att köra funktionen och är angiven i boxarna i funktionskartan.
- Frivilliga – input som inte krävs för köra funktionen men kan användas för att till exempel undvika manuell input.
- Manuella – alternativ som användaren måste välja när funktionen har startat.

Ordningen som inputen är angiven i på funktionskartan är samma som ordningen som krävs för att köra funktionerna. För att tydliggöra finns det för vissa funktioner beskrivningar av variabelnamn och format på inputen, men detta saknas generellt. För användning av datorverktyget rekommenderas att Huvudfunktionen.m startas först, när huvudfunktionen körs kommer datorverktygets samtliga valmöjligheter att visas. Val av funktion och val av alternativ i funktionerna sker sedan genom sifferinmatning.



Appendix C. Exempel på indata

```

function [ Actor ] = Exempel_pa_indata( )
% This function contains the information of the mapped actors
% dependent on
% influencing on

%==== INDATA AKTÖRER =====
% Actor 1-3 är påhittade för testdata

%--- Actor 1 -----
Actor{1}.Name='Aktör 1';
Actor{1}.ExpertNames={'Names1','Name2'};
Actor{1}.Units={'Unit1','Unit2'};
Actor{1}.Titles={'Title1','title2'};
Actor{1}.Goals={'goal1','goal2','goal3'};
Actor{1}.Activities={'Aktivitet1','Aktivitet2'};
Actor{1}.MatrixActGoals=[... %Rader = Activities, kolumner = Goals
4 5 3;
2 2 2];

Actor{1}.NbrInfluencingFlows=3;
Actor{1}.InfFlowName{1}='FlowName1';
Actor{1}.InfFlowType(1)=1;
Actor{1}.InfFlowMatrix{1}=[... %Rader = Activities, kolumner = De olika tidsstegen 0-2h, 2-
8h, osv.
0.5 0 0 0 0 0 0;
0.0 0 0 0 0 0 0];

Actor{1}.InfFlowName{2}='FlowName2';
Actor{1}.InfFlowType(2)=2;
Actor{1}.InfFlowMatrix{2}=[...
0.0 0 0 0 0 0 0;
0.5 0 0 0 0 0 0];

Actor{1}.InfFlowName{3}='FlowName3';
Actor{1}.InfFlowType(3)=3;
Actor{1}.InfFlowMatrix{3}=[...
0.75 0 0 0 0 0 0;
0.0 0 0 0 0 0 0];

Actor{1}.NbrDependentFlows=2;
Actor{1}.DepFlowName{1}='FlowName1';
Actor{1}.DepFlowType(1)=1;
Actor{1}.DepFlowMatrix{1}=[...
0.75 0 0 0 0 0 0;
0.5 0 0 0 0 0 0];

Actor{1}.DepFlowName{2}='FlowName3';
Actor{1}.DepFlowType(2)=3;
Actor{1}.DepFlowMatrix{2}=[...
1 0 0 0 0 0 0;
0.5 0 0 0 0 0 1];

%--- Actor 2 -----
Actor{2}.Name='Aktör 2';
Actor{2}.ExpertNames={'Names1','Name2'};
Actor{2}.Units={'Unit1','Unit2'};
Actor{2}.Titles={'Title1','title2'};
Actor{2}.Goals={'goal1'};
Actor{2}.Activities={'Aktivitet1'};
%Actor{2}.Activities={'Aktivitet1','Aktivitet2','Aktivitet3','Aktivitet4','Aktivitet5','Aktivi
tet6','Aktivitet'};
Actor{2}.MatrixActGoals=[... %Rader = Activities, kolumner = Goals
4];

Actor{2}.NbrInfluencingFlows=1;
Actor{2}.InfFlowName{1}='FlowName1';
Actor{2}.InfFlowType(1)=1;
Actor{2}.InfFlowMatrix{1}=[... %Rader = Activities, kolumner = De olika tidsstegen 0-2h, 2-
8h, osv. (i enlighet med "TimeCategories" i funktionen "VisualizingFlows.m")
0.5 0 0 0 0 0 0];

Actor{2}.NbrDependentFlows=1;
Actor{2}.DepFlowName{1}='FlowName2';
Actor{2}.DepFlowType(1)=2;
Actor{2}.DepFlowMatrix{1}=[...
0.75 0 0 0 0 0 0];

%--- Actor 3 -----
Actor{3}.Name='Aktör 3';
Actor{3}.ExpertNames={'Names1','Name2'};
Actor{3}.Units={'Unit1','Unit2'};
Actor{3}.Titles={'Title1','title2'};
Actor{3}.Goals={'goal1'};

```

```
Actor{3}.Activities={'Aktivitet1'};
Actor{3}.MatrixActGoals=[... %Rader = Activities, Kolumner = Goals
1  ];

Actor{3}.NbrInfluencingFlows=1;
Actor{3}.InfFlowName{1}='FlowName2';
Actor{3}.InfFlowType(1)=2;
Actor{3}.InfFlowMatrix{1}=[... %Rader = Activities, Kolumner = De olika tidsstegen 0-2h, 2-
8h, osv. (i enighet med "TimeCategories" i funktionen "VisualizingFlows.m")
1 0 0 0 0 0 0];

Actor{3}.NbrDependentFlows=1;
Actor{3}.DepFlowName{1}='FlowName3';
Actor{3}.DepFlowType(1)=3;
Actor{3}.DepFlowMatrix{1}=[...
0.5 0 0 0 0 0 0];

end
```

Appendix D. MATLAB-kod för datorverktyget

Innehållsförteckning

Appendix 1.1 Kodens begränsningar	D-2
Appendix 1.2 Huvudfunktionen	D-3
Appendix 1.3 Indatacheck.m	D-4
Appendix 1.4 Figur1_creator.m	D-5
Appendix 1.5 Figur2_creator.m	D-7
Appendix 1.6 Figur3_creator.m	D-9
Appendix 1.7 Figur4_creator.m	D-11
Appendix 1.8 Figur5_creator.m	D-14
Appendix 1.9 Figur6_creator.m	D-17
Appendix 1.10 FigurBowtie_Creator	D-21
Appendix 1.11 Visualizing_indata.m	D-25
Appendix 1.12 Responsgraf.m	D-30
Appendix 1.13 Elimination.m	D-32
Appendix 1.14 CreateMatrix.m	D-34
Appendix 1.15 MinMaxCreator.m	D-35
Appendix 1.16 RG.m	D-36
Appendix 1.17 SystemIteration	D-38

Appendix 1.1 Kodens begränsningar

- Matriserna som generas i `Figur2_creator.m`, `Figur3_creator.m` och `Figur4_creator.m` bildar mindre rutor när fler aktörer och flöden adderas till systemet. Det kan bli svårt att skriva ut dessa bilder med läsbar text. I MATLAB kommer det däremot vara möjligt att förstora bilden.
- De figurer som visualiseras i `Visualizing_indata` går inte att generera för större system än 10 aktörer och aktörerna kan inte ha fler än 8 flöden som den kan påverka eller vara beroende av.
- I datorverktyget saknas det stöd för att hantera aktörer som inte påverkar något flöde.
- I datorverktyget saknas det stöd för att hantera aktörer som inte är beroenden av något flöde.

Appendix 1.2 Huvudfunktionen

```

function [ ] = Huvudfunktionen(PicChoice )% En funktion för att det ska kunna gå att ange
önskad figur direkt
%close all
%Huvudfunktionen ska hålla reda på alla andra funktioner och kalla på
%respektive bild.
FlowType=IndataFlowsLund;
%Actors=IndataActorsLund('Eng'); To get the names in english
Actors=IndataActorsLund;
%Actors=IndataActorsLund('Eng');
t1=1; %Behövs i CreateMatrix men ska hämtas i respektive funktion för dess ändamål....
[~,~,ActorActivities,TotalNrOfActivities]=CreateMatrix(t1,FlowType,Actors); % Hämtar
ActorActivities och TotalNrOfActivities till andra funktioner
if nargin == 0
    disp([char(10),...
        '1: Aktörers "faktiska" beroende av flöden',char(10),...
        '2: Skapar en aktör-flöde matris',char(10),...
        '3: Skapar en flöde-flöde matris',char(10),...
        '4: Skapar en aktör-aktör matris',char(10),...
        '5: Rankar 5 viktigaste aktörerna och flödena för en aktör och kommun',char(10),...
        '6: Visar spridningskedjor från en aktör till alla påverkade aktörer inklusive
kapaciteter',char(10)...
        '7: Spridning- och beroendekedjor för en aktör (bowtie)',char(10),...
        '9: Visualisera indatan',char(10),...
        '10: Responsgraf', char(10),...
        '11: Indatakontroll',char(10),...
        ]);
    PicChoice = input('Välj funktion: ');
end
if PicChoice == 1
    Figur1_creator(FlowType,Actors,ActorActivities,TotalNrOfActivities )
end
if PicChoice==2
    Figur2_creator( FlowType,Actors,TotalNrOfActivities )
end
if PicChoice==3
    Figur3_creator(FlowType,Actors,TotalNrOfActivities )
end
if PicChoice==4
    Figur4_creator( FlowType,Actors,ActorActivities,TotalNrOfActivities )
end
if PicChoice==5
    Figur5_creator( FlowType,Actors,ActorActivities,TotalNrOfActivities );
end
if PicChoice==6
    Figur6_Creator( FlowType,Actors)
end
if PicChoice==7
    FigurBowtie_Creator( FlowType,Actors)
end
if PicChoice == 9
    Visualizing_indata(FlowType,Actors)
end
if PicChoice==10
    Responsgraf(FlowType,Actors)
end
if PicChoice==11
    IndataCheck(FlowType,Actors)
end
PicChoice=[];
end

```

Appendix 1.3 Indatacheck.m

```

function [ ] = IndataCheck(FlowType,Actor )
%UNTITLED Summary of this function goes here
% Funktionen kollar om indatafilerna verkar korrekta. Kontrollen är inte
% på något sätt fullständig utan snarare än kontroll av saker som uppkommit
% under utvecklingen av verktyget
% TODO:
% * Kolla så att flödesnr som angivits i IndataActors även finns med i
% IndataFlows

LegalValues=[0,0.13,0.38,0.63,0.88,1];
for a1=1:length(Actor)
    for a2=1:length(Actor{a1}.DepFlowMatrix)
        Diff=setdiff(Actor{a1}.DepFlowMatrix{a2},LegalValues);
        if ~isempty(Diff)
            disp(['Error in Actor{',num2str(a1), '}.DepFlowMatrix{',num2str(a2), '}. Diff:']
num2str(Diff)])
        end
    end
    for a2=1:length(Actor{a1}.InfFlowMatrix)
        Diff=setdiff(Actor{a1}.InfFlowMatrix{a2},LegalValues);
        if ~isempty(Diff)
            disp(['Error in Actor{',num2str(a1), '}.InfFlowMatrix{',num2str(a2), '}. Diff:']
num2str(Diff)])
        end
    end
    if ~length(Actor{a1}.Activities) == size(Actor{a1}.MatrixActGoals,1)
        disp(['Error in Actor{',num2str(a1), '}.Activities}. or MatrixActGoals'])
    end
    if ~length(Actor{a1}.Goals) == size(Actor{a1}.MatrixActGoals,2)
        disp(['Error in Actor{',num2str(a1), '}.Activities}. or MatrixActGoals'])
    end
    if ~Actor{a1}.NbrInfluencingFlows==length(Actor{a1}.InfFlowMatrix)
        disp(['Error in Actor{',num2str(a1), '}.NbrInfluencingFlows} or
length(Actor{a1}.InfFlowMatrix)'])
    end
    if ~Actor{a1}.NbrDependentFlows==length(Actor{a1}.DepFlowMatrix)
        disp(['Error in Actor{',num2str(a1), '}.NbrInfluencingFlows} or
length(Actor{a1}.InfFlowMatrix)'])
    end
end
% FlowType; % får inte var kortare än högsta numret på:
% Actor{a1}.InfFlowType(1); %ochh
% Actor{a1}.DepFlowType(1);
% Actor{1}.InfFlowMatrix; %Rader = Activities, Kolumner = De olika tidsstegen 0-2h, 2-8h, osv.
(i enlighet med "TimeCategories" i funktionen "VisualizingFlows.m")
end

```

Appendix 1.4 Figur1_creator.m

```

function [ ] = Figur1_creator(FlowType,Actors,ActorActivities,TotalNrOfActivities )

%% Skapar en figur med flera subplottar. Varje subplot representerar ett
% flöde och trappstegen i figuren visar hur varje aktör blir påverkad över
% tiden om ett flöde blir fullständigt utslaget. Endast de flöde som blir
% påverkade ritas ut.
% Här finns inga valmöjligheter, alla aktörer, alla flöden och alla
% tidsperspektiv studeras.

%% För att göra om ActivitiesCapacity till ActorCapacity behövs följande del
%ActivitiesGoal är samma oavsätt tidssteg så den behöver endast räknas ut en gång
TotGoalsum=zeros(1,length(Actors)); % Räknar ut respektive aktörs
% totala målsomma.
for a1=1:length(Actors)
    TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
end

ActivitiesGoal=zeros(1,TotalNrOfActivities); % Räknar ut hur stor betydelse respektive
aktivitet har för aktören
a3=0;
for a1=1:length(Actors)
    for a2=1:length(Actors{a1}.Activities)
        [~,ActorCol]=find(ActorActivities == a1);
        for a3=1:length(ActorCol)
            ActivitiesGoal(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1);
        end
    end
end

%% Skapar en 3d matris med aktörers beroende av flöden i de 7 tidsintervallen
%Samtliga flöden blir fullständigt utslagna en gång för varje tidsintervall
ActorDepOnFlows=zeros(length(Actors),length(FlowType),7);
for t1=1:7 % Hämtar data för de sju tidsintervallen
    ActivitiesDepOnFlow=zeros(TotalNrOfActivities,length(FlowType)); % Skapar en matris som
    används senare, måste nollas i varje tidsloop.
    for q1=1:length(FlowType)
        InoperableActivities=zeros(1,TotalNrOfActivities); % Alla aktiviteter är 100% i början
        på varje loop
        InoperableFlows=zeros(length(FlowType),1); % Nollställer från föregående utslagning
        InoperableFlows(q1)=1; % Slår ut ett flöde fullständigt åt gången
        [ActivitiesCapacity,~,ActorActivities,TotalNrOfActivities,~,~] =
        SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows); % Kör själva koden
        (SystemIteration) med de utslagna flödena
        ActivitiesDepOnFlow(:,q1)=ActivitiesCapacity(:); %Sparar effekten av en utslagning av
        ett flöde
    end
    ActivitiesDepOnFlow=1-ActivitiesDepOnFlow; % gör om kapacitet till nedsättning

    % Gör om flöde-aktivitet till flöde-aktör med hjälp av hur "viktiga"
    % respektive aktivitet är för aktören.
    for a1=1:length(Actors)
        for a2=1:length(FlowType)
            for a3=1:TotalNrOfActivities
                if a1==ActorActivities(a3)

ActorDepOnFlows(a1,a2,t1)=ActorDepOnFlows(a1,a2,t1)+ActivitiesDepOnFlow(a3,a2)*ActivitiesGoal(
a3);
                end
            end
        end
    end
end

%% Går igenom indatan och ser om där finns flöden som inte någon aktör inte
% är beroende av.
ExistingDepOnFlows=ones(1,length(FlowType));
for a1=1:length(Actors)
    for a2=1:length(FlowType)
        for a3=1:7
            if ActorDepOnFlows(:,a2,')==0 % Kontrollerar om samtliga aktörer under alla sju
tidsstegen är oberoende av flödet.
                ExistingDepOnFlows(a2)=0;
            end
        end
    end
end
CountExistingDepOnFlows=sum(ExistingDepOnFlows); % Gör en snabb kontroll över hur många
flöden någon aktör är beroende av
for L1=1:length(Actors)
    ActorsNrAndName{L1} = [ num2str(L1) ' : ' char([Actors{L1}.Name)) char(10)]; % Skapar en
variabel med alla nummer och namn på aktörerna
end

%% --- Visualisering -----
%Med följande tidskategorier

```

```

TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'Veckor';'Månader';'År'};
% Skapar ett färgschema som går från rött till rött med hsv funktionen
color=zeros(length(Actors),3);
for a2=1:length(Actors)
    color(:,:)=hsv(length(Actors));
end

%% Skriver ut alla flöden som ej används
a1=0;
for a=1:length(ExistingDepOnFlows)
    if ExistingDepOnFlows(a)==0
        for a1=a1+1
            FlowName{a1} = [num2str(a) ' : ' char(FlowType(a)), char(10)]; %Skapar en variabel
            med namn och printar en per rad
        end
    end
end
disp(['Följande flöden är ingen aktör beroende av',char(10), strjoin(FlowName)])

%% Skapar figurer med (11 subplottar med flöden)/figur som någon aktör är beroende av
i=0;
ag = findobj; % all graphical objects
nf = max(ag(ag==fix(ag))); % maximal integer value is number of figures
q1=nf+1;
for a1=1:length(FlowType)
    figure(q1)
    set(gcf, 'Position',[250 80 1000 720]);
    if i<=10 % Om där är fler än 11 flöde som aktörerna är beroende av avbryts figur1
        if ExistingDepOnFlows(a1)==1 % Tittar bara på de flödena som är aktuella
            i=i+1;
            subplot(6,2,i) % Sätter ut subplotten på rätt ställe.
            title(FlowType(a1), 'fontweight', 'bold')
            hold on
            for a3=1:7
                for a2=1:length(Actors)
                    if mod(a2,2)==1 % Ritar ut trappsteg med varanan aktör sträckad och
                        ActorDepOnFlows(a2,a1,end)]+0.02*rand, '-', 'Color', color(a2,:), 'Linewidth', 2) %Transpose och
                        squeeze "används för att lägga matrisen på rätt håll"
                    else
                        ActorDepOnFlows(a2,a1,end)]+0.02*rand, '--', 'Color', color(a2,:), 'Linewidth', 2)
                    end
                end
            end
            axis([1 8 0 1.1]) % Sätter rätt axlar på subplottarna, tidsstegen och till
            vilken påverkans grad.
            set(gca, 'xtick', [1.5:1:7.5])
            set(gca, 'xtickLabel', TimeCategories)
            ylabel('Störning')
            xlabel('Tidsintervall')
            box on
        end
    end
else
    ML=legend(ActorsNrAndName); %Skapar en legend
    newPosition = [0.7 0 0.1 0.2];
    set(ML, 'Position', newPosition); % Flyttar legenden så att den ligger brevid
    i=0; % Nollställer positionen för subplottarna
    q1=q1+1; % Andrar till nästa figur när villkoret är uppfyllt
end
end
% Skriver ut legenden för den sista matrisen med.
ML=legend(ActorsNrAndName); %Skapar en legend
newPosition = [0.7 0 0.1 0.2];
set(ML, 'Position', newPosition); % Flyttar legenden så att den ligger brevid
end

```


Appendix 1.5 Figur2_creator.m

```

function [ ] = Figur2_creator(FlowType,Actors,TotalNrOfActivities )
% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet, matrisen går sedan vidare och med hjälp av de olika
% aktörernas mål görs matrisen om till aktör-aktör. Avslutningsvis ritas en
% figur med färgade rutor med siffror som beskriver hur starkt beroende
% olika aktörer har av andra aktörer (Hur de faktiskt blir påverkade om en
% andra aktörer blir fullständigt utslagna).
t1 = input('Välj tidssteg: ');
LenActor=length(Actors);
% Skapar matriser som endast behöver skapas en gång.
farg=zeros(length(FlowType),length(FlowType),3);
%% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet -> aktivitet-aktör matris
for q1=1:length(FlowType)
    InoperableActivities=zeros(1,TotalNrOfActivities); % Samtliga aktiviteter är opåverkade
    till en början.
    InoperableFlows=zeros(length(FlowType),1); % Samtliga flöde är opåverkade till en början.
    InoperableFlows(q1)=1;
    [ActivitiesCapacity,~,ActorActivities,~,~,~] =
    SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows);
    ActivitiesDepOnFlows(q1,:)=ActivitiesCapacity(:); % Sparar effekten av en utslagning av en
    aktör. Visar statusen för varje aktivitet när aktören (RadNR) är påverkad
end
ActivitiesDepOnFlows=1-transpose(ActivitiesDepOnFlows); % Gör om kapacitet till nedsättning.

%% Skapar en vektor med hur viktiga aktiviteterna är för respektive aktör.
% Skapar en vektor med hur många "målpoäng" varje aktör har
TotGoalsum=zeros(1,LenActor);
for a1=1:LenActor
    TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
end
% Skapar en vektor med hur "viktig" varje aktivitet är för aktörens måluppyllnad
ActivitiesGoal=zeros(1,TotalNrOfActivities);
a3=0;
for a1=1:LenActor
    for a2=1:length(Actors{a1}.Activities)
        [~,ActorCol]=find(ActorActivities == a1);
        for a3=1:length(ActorCol)
            ActivitiesGoal(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1);
        end
    end
end
end

%% Gör om aktivitet-aktivitet till aktör-aktör
% I omvandling görs antagandet att alla aktiviteter tillsammans bygger upp aktörens kapacitet
ActorDepOnFlows=zeros(LenActor,length(FlowType));
for a1=1:LenActor
    for a2=1:length(FlowType)
        for a3=1:TotalNrOfActivities
            if a1==ActorActivities(a3)

ActorDepOnFlows(a1,a2)=ActorDepOnFlows(a1,a2)+ActivitiesDepOnFlows(a3,a2)*ActivitiesGoal(a3);
%
            end
        end
    end
end

%% Skapar en färgglad matris
for a1=1:LenActor
    for a2=1:length(FlowType)
        temp=RG(ActorDepOnFlows(a1,a2)); % Hämtar färgkoden motsvarande kapaciteten.
        for a3=1:3
            farg(a1,a2,a3)=temp(a3); % Sparar färgkoderna i en tredje dimension för respektive
            aktör-aktör beroende
        end
    end
end
F=figure;
set(F,'units','normalized','outerposition',[0 0 1 1])
image(0.5,0.5,farg)

%% Skriver ut siffran i matrisen
for a1=1:LenActor
    for a2=1:length(FlowType)
        ActorDepOnFlowNumbers{a1,a2} = num2str(ActorDepOnFlows(a1,a2),2); % Hämtar
        beroendegraden med 2 decimaler och gör om den till string från dubbel.
    end
end
%ActorDepOnFlowNumbers1=transpose(ActorDepOnFlowNumbers); % Ändrar så att den skriver ut
siffran i rätt ruta.
for a1=1:LenActor
    Y(1,a1)=a1;
end

```

```

for a1=1:length(FlowType)
    X(1,a1)=a1;
end
X=X-0.5;
Y=Y-0.6;
for a3=1:LenActor
    for a4=1:length(FlowType)

text(X(a4),Y(a3),ActorDepOnFlowNumbers(a3,a4),'HorizontalAlignment','center','VerticalAlignmen
t','top')
end
end
X=X+0.5;
Y=Y+0.6;

%% Skriver ut namnen till vänster om matrisen
X=X-LenActor-0.1;
Y=Y-0.6;
for a4=1:LenActor
    text(X(a3),Y(a4),Actors{a4}.Name,'HorizontalAlignment','right','VerticalAlignment','top')
end
X=X+LenActor+0.1;
Y=Y+0.6;

%% Skriver ut namnen snett över matrisen
X=X-0.5;
Y=Y-LenActor-0.1;
for a4=1:length(FlowType)

text(X(a4),Y(LenActor),FlowType(a4),'HorizontalAlignment','left','VerticalAlignment','bottom',
'rotation',45)
end
axis([-1 length(FlowType) -1 LenActor])% Skapar -1 marginalen för att namnen ska skrivas och
vara möjliga att läsa.
axis off
grid minor
set(gca,'xticklabel',[],'ytick',[]);
%set(gcf,'units','normalized','position',[0 .1 .8 0.8])
%set(gca,'units','normalized','position',[0.15 .15 .75 0.75])
%set(gca,'xtick',[],'ytick',[]);

%% Titlar
text(-0.9*(LenActor)/2,(LenActor)/2,'Påverkade
aktörer','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',90,'fontweight
','bold')
text((length(FlowType)/2),-0.27*(LenActor)/2,'Utslagna
flöden','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',0,'fontweight',
'bold')
TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'veckor';'Månader';'År'};
text(length(FlowType)/2,1.05*LenActor,['Tidsperid:
',char(TimeCategories{t1})],'HorizontalAlignment','center','VerticalAlignment','bottom','rotat
ion',0,'fontweight','bold')

end

```

Appendix 1.6 Figur3_creator.m

```

function [ ] = Figur3_creator(FlowType,Actors,TotalNrOfActivities )
% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet, matrisen går sedan vidare och med hjälp av de olika
% aktörernas mål görs matrisen om till aktör-aktör. Avslutningsvis ritas en
% figur med färgade rutor med siffror som beskriver hur starkt beroende
% olika aktörer har av andra aktörer (Hur de faktiskt blir påverkade om en
% andra aktörer blir fullständigt utslagna).
t1 = input('Välj tidssteg: ');

% Skapar matriser som endast behöver skapas en gång.
farg=zeros(length(FlowType),length(FlowType),3);
%% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet -> aktivitet-aktör matris
for q1=1:length(FlowType)
    InoperableActivities=zeros(1,TotalNrOfActivities); % Samtliga aktiviteter är opåverkade
    till en början.
    InoperableFlows=zeros(length(FlowType),1); % Samtliga flöde är opåverkade till en början.
    InoperableFlows(q1)=1;
    [~,FlowCapacity,~,~,~,~] =
SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows);
    FlowDepOnFlows(:,q1)=FlowCapacity(:); % Sparar effekten av en utslagning av en aktör.
    visar statusen för varje aktivitet när aktören (RadNR) är påverkad
end
FlowDepOnFlows=1-FlowDepOnFlows; % Gör om kapacitet till nedsättning.
FlowDepOnFlows(eye(length(FlowType))==1)=1.0001;

%% Skapar en färgglad matris
for a1=1:length(FlowType)
    for a2=1:length(FlowType)
        temp=RG(FlowDepOnFlows(a1,a2)); % Hämtar färgkoden motsvarande kapaciteten.
        for a3=1:3
            farg(a1,a2,a3)=temp(a3); % Sparar färgkoderna i en tredje dimension för respektive
            aktör-aktör beroende
        end
    end
end
F=figure;
set(F,'units','normalized','position',[0 .2 .7 .7])
image(0.5,0.5,farg)

%% Skriver ut siffran i matrisen
for a1=1:length(FlowType)
    for a2=1:length(FlowType)
        FlowNumbers{a1,a2} = num2str(FlowDepOnFlows(a1,a2),2); % Hämtar beroendegraden med 2
        decimaler och gör om den till string från dubbel.
    end
end
FlowNumbers1=transpose(FlowNumbers); % Ändrar så att den skriver ut siffran i rätt ruta.
for a1=1:length(FlowType)
    X(1,a1)=a1;
    Y(1,a1)=a1;
end
X=X-0.5;
Y=Y-0.6;
for a3=1:length(FlowType)
    for a4=1:length(FlowType)
        text(X(a3),Y(a4),FlowNumbers1(a3,a4),'HorizontalAlignment','center','VerticalAlignment','top')
    end
end
X=X+0.5;
Y=Y+0.6;

%% Skriver ut namnen till vänster om matrisen
X=X-length(FlowType)-0.1;
Y=Y-0.7;
for a4=1:length(FlowType)
    text(X(a3),Y(a4),FlowType(a4),'HorizontalAlignment','right','VerticalAlignment','top')
end
X=X+length(FlowType)+0.1;
Y=Y+0.7;

%% Skriver ut namnen snett över matrisen
X=X-0.5;
Y=Y-length(FlowType)-0.1;
for a4=1:length(FlowType)
    text(X(a4),Y(a1),FlowType(a4),'HorizontalAlignment','left','VerticalAlignment','bottom','rotation',45)
end

axis([-1 length(FlowType) -1 length(FlowType)]) % Skapar -1 marginalen för att namnen ska
skrivas och vara möjliga att läsa.
axis on
grid minor

```

```
set(gca,'xticklabel',[],'yticklabel',[]);
set(gca,'xtick',[],'ytick',[]);
axis off
text(-3.9,(length(FlowType)/2),'Påverkade
flöden','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',90,'fontweight'
,'bold')
text(0.5*(length(FlowType)),-0.23*(length(FlowType)/2),'Utslagna
flöden','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',0,'fontweight'
,'bold')
TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'Veckor';'Månader';'År'};
text(0.5*(length(FlowType)),1.1*max(X),['Tidsperiod:
',char(TimeCategories{t1})],'HorizontalAlignment','center','VerticalAlignment','bottom','rotat
ion',0,'fontweight','bold')
end
```

Appendix 1.7 Figur4_creator.m

```

function [ ] = Figur4_creator(FlowType,Actors,ActorActivities,TotalNrOfActivities)
% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet, matrisen går sedan vidare och med hjälp av de olika
% aktörernas mål görs matrisen om till aktör-aktör. Avslutningsvis ritas en
% figur med färgade rutor med siffror som beskriver hur starkt beroende
% olika aktörer har av andra aktörer (Hur de faktiskt blir påverkade om en
% andra aktörer blir fullständigt utslagna). Den första figuren antar
% medelvärdet för spridningsintervallet och de tre mindre figurena till
% höger antar min, max och den sista beskriver differensen mellan dem.
if nargin < 5 % Kör vanliga medelvärden när inget annats angetts
    MinMax = [];
end
t1 = input('Välj tidssteg: ');
% Skapar matriser som endast behöver skapas en gång.
ActivitiesDepOnActors=zeros(TotalNrOfActivities,length(Actors));
ActorsDepOnActivities=zeros(length(Actors),TotalNrOfActivities); % Skapar en matris som
används senare
InoperableActors=ones(1,TotalNrOfActivities);
farg=zeros(length(Actors),length(Actors),3);

%% Slår ut en aktör i taget fullständigt och sparar det i en matris
% aktör-aktivitet -> aktivitet-aktör matris
MinMax={'min','medel','max'};
for c1=1:3
    for q1=1:length(Actors)
        InoperableActivities=zeros(1,TotalNrOfActivities);
        [~,ActiveCol]=find(ActorActivities == q1); %ta reda på alla positioner i
        ActorActivities som innehåller q1, dvs
        %numret på den utslagna aktören
        InoperableActivities(ActiveCol)=InoperableActors(ActiveCol); % Slår ut aktörens
        aktiviteter.
        InoperableFlows=zeros(length(FlowType),1); % Samtliga flöde är opåverkade till en
        början.
        [ActivitiesCapacity,~,ActorActivities,~,~,~] =
        SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows, MinMax(c1));
        ActorsDepOnActivities(q1,:)=transpose(ActivitiesCapacity(:)); % Sparar effekten av en
        utslagning av en aktör. Visar statusen för varje aktivitet när aktören (RadNR) är påverkad
        end
        ActivitiesDepOnActors=transpose((1-ActorsDepOnActivities)); % Gör om kapacitet till
        nedsättning.

        %% Skapar en vektor med hur viktiga aktiviteterna är för respektive aktör.
        % Skapar en vektor med hur många "målpoäng" varje aktör har
        TotGoalsum=zeros(1,length(Actors));
        for a1=1:length(Actors)
            TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
        end
        % Skapar en vektor med hur "viktig" varje aktivitet är för aktörens måluppyllnad
        ActivitiesGoal=zeros(1,TotalNrOfActivities);
        a3=0;
        for a1=1:length(Actors)
            for a2=1:length(Actors{a1}.Activities)
                [~,ActorCol]=find(ActorActivities == a1);
                for a3=1:length(ActorCol)
                    ActivitiesGoal(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1); % Summerar
                    aktivitetens "målpoäng" och delar med aktörens totala "målpoäng"
                end
            end
        end

        %% Gör om aktivitet-aktivitet till aktör-aktör
        % I omvandling görs antagandet att alla aktiviteter tillsammans bygger upp aktörens
        kapacitet
        ActorDepOnActorNum=zeros(length(Actors),length(Actors));
        for a1=1:length(Actors)
            for a2=1:length(Actors)
                for a3=1:TotalNrOfActivities
                    if a1==ActorActivities(a3)
                        ActorDepOnActorNum(a1,a2)=ActorDepOnActorNum(a1,a2)+ActivitiesDepOnActors(a3,a2)*ActivitiesGoa
                        l(a3); %
                    end
                end
            end
        end

        ActorDepOnActorNum(a1,a2)=ActorDepOnActorNum(a1,a2)+1.0001; % För att skapa gråa rutor (se även
        RG funktionen)
    end
end
end
end
if c1==1
    minvalues=ActorDepOnActorNum; % Sparar beorendegraden för de minsta värdena i
    spridningsintervallen
end
if c1==3

```

```

        maxvalues=ActorDepOnActorNum; % Sparar beorendegraden för de största värdena i
spridningsintervallen
        Diffvalues=maxvalues-minvalues; %Sparar diffensen mellan max och min värdena
    end
end
ActorDepOnActor=ActorDepOnActorNum;

%% Skapar en färgglad matris
for a1=1:length(Actors)
    for a2=1:length(Actors)
        temp=RG(ActorDepOnActor(a1,a2)); % Hämtar färgkoden motsvarande kapaciteten.
        for a3=1:3
            farg(a1,a2,a3)=temp(a3); % Sparar färgkoderna i en tredje dimension för respektive
aktör-aktör beroende
        end
    end
end
figure %För den stora matrisen
subplot(3,4,[1 2 3 5 6 7 9 10 11]) %Beskriver den stora matrisens position
image(0.5,0.5,farg) %Färglägger bilden.

%% Skriver ut siffran i matrisen
for a1=1:length(Actors)
    for a2=1:length(Actors)
        ActorNumbers{a1,a2} = num2str(ActorDepOnActor(a1,a2),2); % Hämtar beroendegraden med 2
decimaler och gör om den till string från dubbel.
    end
end
ActorNumbers1=transpose(ActorNumbers); % Ändrar så att den skriver ut siffran i rätt ruta.
for a1=1:length(Actors)
    X(1,a1)=a1;
    Y(1,a1)=a1;
end
X=X-0.5;
Y=Y-0.5;
for a3=1:length(Actors)
    for a4=1:length(Actors)

text(X(a3),Y(a4),ActorNumbers1(a3,a4),'HorizontalAlignment','center','VerticalAlignment','top'
,'FontSize',12)
    end
end
X=X+0.5;
Y=Y+0.5;
%% Skriver ut namnen till höger om matrisen
X=X-length(Actors)-0.1;
Y=Y-0.5;
for a4=1:length(Actors)

text(X(a3),Y(a4),Actors{a4}.Name,'HorizontalAlignment','right','VerticalAlignment','top','Font
Size',12)
end
X=X+length(Actors)+0.1;
Y=Y+0.5;
%% Skriver ut namnen snett över matrisen
X=X-0.5;
Y=Y-length(Actors)-0.1;
for a4=1:length(Actors)

text(X(a4),Y(a1),Actors{a4}.Name,'HorizontalAlignment','left','VerticalAlignment','bottom','ro
tation',45,'FontSize',12)
end

axis([-1 length(Actors) -1 length(Actors)]) % Skapar -1 marginalen för att namnen ska skrivas
och vara möjliga att läsa.
axis off
grid on
set(gca,'xticklabel',[],'yticklabel',[]);
text((-length(Actors)/3.1),(length(Actors)/2),'Påverkade
aktörer','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',90,'fontweight
','bold')
text(0.5*(length(Actors)),-0.3*(length(Actors)/1.6),'Utslagna
aktörer','HorizontalAlignment','center','VerticalAlignment','bottom','rotation',0,'fontweight'
,'bold')
TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'Veckor';'Månader';'År'};
text(0.5*(length(Actors)),1.15*max(X),['Tidsperid:
','char(TimeCategories{t1})'],'HorizontalAlignment','center','VerticalAlignment','bottom','rotat
ion',0,'fontweight','bold')

%% Skapar datan till de små max och min matriserna till höger.
for a1=1:length(Actors)
    for a2=1:length(Actors)
        temp1=RG(minvalues(a1,a2)); % Hämtar färgkoden motsvarande kapaciteten.
        for a3=1:3
            fargMin(a1,a2,a3)=temp1(a3); % Sparar färgkoderna i en tredje dimension för
respektive aktör-aktör beroende
        end
    end
end

```

```

    end
end
for a1=1:length(Actors)
    for a2=1:length(Actors)
        temp2=RG(maxvalues(a1,a2)); % Hämtar färgkoden motsvarande kapaciteten.
        for a3=1:3
            fargMax(a1,a2,a3)=temp2(a3); % Sparar färgkoderna i en tredje dimension för
respektive aktör-aktör beroende
        end
    end
end
DiffColours=zeros(length(Actors),length(Actors),3);

%% Skapar datan till diffmatrisen till höger.
GreyEye=eye(length(Actors));
Diffvalues(GreyEye==1)=1.0001;
for a1=1:length(Actors)
    for a2=1:length(Actors)
        if Diffvalues(a1,a2) == 1.0001
            DiffColours(a1,a2,:)=RG(1.0001);
        elseif Diffvalues(a1,a2)>0.25
            DiffColours(a1,a2,:)=RG(0.99);
        elseif Diffvalues(a1,a2)>0.13
            DiffColours(a1,a2,:)=RG(0.5);
        elseif Diffvalues(a1,a2)<=0.13
            DiffColours(a1,a2,:)=RG(0.0);
        end
    end
end

%% Ritar ut max och min figurerna till höger
subplot(3,4,4)
image(0.5,0.5,fargMin)
text(0.5*(length(Actors)), -0.1*(length(Actors)/2), 'Min
påverkan', 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', 'rotation', 0, 'fontweight
', 'bold')
axis on
grid on
set(gca, 'xtick', 1:length(Actors), 'ytick', 1:length(Actors));
S2=subplot(3,4,8);
set(S2, 'Tag', 'surf'); % Inklippta
image(0.5,0.5,fargMax)
text(0.5*(length(Actors)), -0.1*(length(Actors)/2), 'Max
påverkan', 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', 'rotation', 0, 'fontweight
', 'bold')

%% Skapar en legend för påverkan
custom_colormap1 = [ RG(0);RG(0.25); RG(0.5);RG(0.75); RG(1);];
custom_colormap2 = [ RG(0); RG(0.5); RG(1);];
custom_colormap=[custom_colormap2];
colormap(custom_colormap);
Labels={'0%', '50%', '100%'};
AxisPos=get(gca, 'position');
BarPos=[AxisPos(1)+AxisPos(3)+0.04, AxisPos(2)];
hcb=colorbar('peer', gca, [BarPos 0.015
0.1], 'YTickLabel', Labels, 'XAxisLocation', 'bottom', 'TickLength', [0 0], 'Color', [1 0 1]);
title(hcb, ['Färgskala påverkan', char(10), '%'], 'HorizontalAlignment', 'center');

%% Ritar ut diff figuren till höger
axis on
grid on
set(gca, 'xtick', 1:length(Actors), 'ytick', 1:length(Actors));
S3=subplot(3,4,12);
image(0.5,0.5,DiffColours)
axis on
grid on
set(gca, 'xtick', 1:length(Actors), 'ytick', 1:length(Actors));
text(0.5*(length(Actors)), -0.1*(length(Actors)/2), 'Differens max &
min', 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom', 'rotation', 0, 'fontweight', 'bo
ld')

%% Skapar en legend för diffen
colormap(custom_colormap2);
Labels={'<13%', '13-25%', '>25%'};
AxisPos=get(gca, 'position');
BarPos=[AxisPos(1)+AxisPos(3)+0.04, AxisPos(2)];
hcb2=colorbar('peer', gca, [BarPos 0.015
0.1], 'YTickLabel', Labels, 'XAxisLocation', 'bottom', 'TickLength', [0 0], 'Color', [1 0
1]);%,'Position', 'best');
title(hcb2, ['Färgskala differens ', char(10), '%-enheter'], 'HorizontalAlignment', 'center');

end

```

Appendix 1.8 Figur5_creator.m

```

function [ImportanceCity] =
Figur5_creator(FlowType,Actors,ActorActivities,TotalNrOfActivities,Reduction,t1,IntrestingActo
r)
% Skapar figurer för hur viktiga olika aktörer är för kommunen och för en
% enskild (utvald) aktör. Rankningen är baserad på att en simulering körs
% för varje aktör och en 50% utslagning används
LenActors=length(Actors);
ActorsDepOnActivities=zeros(LenActors,TotalNrOfActivities); % Skapar en matris som används
senare
FlowDepOnActivities=zeros(length(FlowType),TotalNrOfActivities);

if nargin <= 4 % för att kunna köra funktionen via responsgrafan, frågar bara om det inte fått
input från kallande funktion
    Reduction=0.5;
    ImportanceToActor=1:1:LenActors; %En rad med ID för alla aktörer
    %% input
    t1 = input('Välj tidssteg:');
    %ActorsNames=cell(LenActors,1);
    for a=1:LenActors
        ActorsNrAndName{a} = [ num2str(a) ' : ' char(Actors{a}.Name) char(10)]; % Variabel med
alla namn och nummer och printar en per rad
    end
    disp(['Möjliga aktörer:' ,char(10), strjoin(ActorsNrAndName)])
    IntrestingActor = input('Välj aktör som det ska visas rankingen för (Lämna tom för enbart
kommunen): ');
    %%
end
InoperableActorsActivities=Reduction*ones(1,TotalNrOfActivities); % Skapar en ActorsReduction
i utslagning på alla aktörer
for q1=1:LenActors
    InoperableActivities=zeros(1,TotalNrOfActivities);
    %ta reda på alla positioner i ActorActivities som innehåller q1, dvs
    %numret på den utslagna aktören
    [~,ActiveCol]=find(ActorActivities == q1);
    InoperableActivities(ActiveCol)=InoperableActorsActivities(ActiveCol); % Tar nedsättningen
från aktuell aktör aktiviter och lagrar dessa.
    InoperableFlows=zeros(length(FlowType),1); % Inga flöden är utslagna

    [ActivitiesCapacity,~,ActorActivities,TotalNrOfActivities,~,~] =
SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows);
    ActorsDepOnActivities(q1,:)=transpose(ActivitiesCapacity(:)); % Sparar effekten av en
utslagning av en aktör. Visar statusen för varje aktivitet när aktören (RadNR) är påverkad
end

ActivitiesDepOnActors=transpose((1-ActorsDepOnActivities)); % E: användes för att förstå
nedstående. Gör om kapacitet till nedsättning.

%% Skapar en vektor med hur många "målpoäng" varje aktör har
TotGoalsum=zeros(1,LenActors);
for a1=1:LenActors
    TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
end
% Skapar en vektor med hur "viktig" varje aktivitet är för aktörens måluppyllnad
ActivitiesGoalWeight=zeros(1,TotalNrOfActivities);
for a1=1:LenActors
    for a2=1:length(Actors{a1}.Activities)
        [~,ActorCol]=find(ActorActivities == a1);
        for a3=1:length(ActorCol)
            ActivitiesGoalWeight(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1);
        end
    end
end

%% Gör om aktör-aktivitet matris till aktör-aktör matris
% Aktiviterna blir till aktörer beroende på hur "viktiga" de är för aktören.
ActorDepOnActor=zeros(LenActors,LenActors);
for a1=1:LenActors
    for a2=1:LenActors
        for a3=1:TotalNrOfActivities
            if a1==ActorActivities(a3)
                ActorDepOnActor(a1,a2)=ActorDepOnActor(a1,a2)+ActivitiesDepOnActors(a3,a2)*ActivitiesGoalweigh
t(a3); % kolumn utslagen och rad visar hur starkt beroendet är,
            end
        end
    end
end

%% Gör om Flöde-aktivitet matris till Flöde-aktör matris
% Aktiviterna blir till aktörer beroende på hur "viktiga" de är för aktören.
FlowDepOnActor=zeros(length(FlowType),LenActors);
for a1=1:length(FlowType)
    for a2=1:LenActors
        for a3=1:TotalNrOfActivities

```



```

        if a2==ActorActivities(a3)
FlowDepOnActor(a1,a2)=FlowDepOnActor(a1,a2)+FlowDepOnActivities(a1,a3)*ActivitiesGoalWeight(a3
); % Kolumn utslagen och rad visar hur starkt beroendet är,
        end
    end
end

%% Skapar vektor som visar aktörernas viktighet för kommunen.
ImportanceCity=1:1:LenActors;
RemoveInitial=eye(LenActors,LenActors);
RemoveInitial(RemoveInitial==1)=RemoveInitial(RemoveInitial==1)-(1+Reduction);
ActorDepOnActor=ActorDepOnActor+RemoveInitial; % Kunde lika bra gjorts på summan, men kanske
är bra om man nu vill kolla aktör-aktör
for a1=1:LenActors
    ImportanceCity(2,a1)=sum(ActorDepOnActor(:,a1)); % viktigheten kan alltså vara [0-Antalet
aktörer]
end

ImportanceCity=transpose(ImportanceCity); %För att sortera, går säkert ändå men lättare att
förstå :)
[~, idx] = sort(ImportanceCity(:,2)); %Tar reda på hur det ska sorteras.
ImportanceCityRanked = flipud(ImportanceCity(idx,:));
ImportanceCityRanked = ImportanceCityRanked(1:5,:); % Tar fram 5 viktigaste
ImportanceCityRanked(:,2)=int64(100.*ImportanceCityRanked(:,2)/LenActors); % För att visa
faktor (0-100) och i heltal
ImportanceCityRanked=num2cell(ImportanceCityRanked);
for a=1:5 % Namn på 5 viktigaste
    ImportanceCityRanked{a,1}=Actors{ImportanceCityRanked{a,1}}.Name;
end
% Skapar tabell och text
f = figure;
set(f,'units','normalized','position',[0 .2 .6 .6]) % Gör figuren lite större så att texten
inte överlappar
colnames = {'Aktör','Beroendefaktor [0-100]'};

XStart=0.05;
YStart=0.5;
XStartLow=0.05;
YStartLow=0.05;
XStep=0.4;
YStep=0.3;

t = uitable(f, 'Columnwidth',{130 140},'Data', ImportanceCityRanked, 'ColumnName', colnames,
...
'units','normalized', 'pos',[XStart YStart XStep YStep]);
uicontrol('style','text','units','normalized','pos',[XStart YStart+YStep XStep
0.1],'stri',{'kommunens beroende av aktörer'},'fonts',10 );

%% Skapar vektor som visar aktörernas viktighet för specifik aktör. Samma kod som i föregående
avsnitt
if ~isempty(IntrestingActor)
    ImportanceToActor(2,:)=ActorDepOnActor(IntrestingActor,:);
    ImportanceToActor=transpose(ImportanceToActor); %För att sortera, går kanske ändå men
lättare att förstå :)
    ImportanceToActor(IntrestingActor,:)=[]; % Tar bort den egna påverkan
    [~, idx] = sort(ImportanceToActor(:,2)); %Tar reda på hur det ska sorteras. Nedanstående i
identiskt med kommunrankningen.
    ImportanceToActorRanked = flipud(ImportanceToActor(idx,:));
    ImportanceToActorRanked = ImportanceToActorRanked(1:5,:);
    ImportanceToActorRanked(:,2)=int64(100.*ImportanceToActorRanked(:,2));
    ImportanceToActorRanked=num2cell(ImportanceToActorRanked);

    for a=1:5
        ImportanceToActorRanked{a,1}=Actors{ImportanceToActorRanked{a,1}}.Name;
    end
    % Skapar tabell och text
    t2 = uitable(f,'columnwidth',{130 140}, 'Data', ImportanceToActorRanked, 'ColumnName',
colnames, ...
'units','normalized', ...
'pos',[XStart+XStep+0.05 YStart XStep YStep]);
    uicontrol('style','text','units','normalized','pos',[XStart+XStep+0.05 YStart+YStep XStep
0.1],'stri',{'Actors{IntrestingActor}.Name','beroende av övriga aktörer'},'fonts',10 );
end

%% Flöde (Liknande det som görs ovan)
ActivitiesAfterFlowInf=zeros(length(FlowType),TotalNrOfActivities);
for q1=1:length(FlowType)
    InoperableActivities=zeros(1,TotalNrOfActivities);
    InoperableFlows=zeros(length(FlowType),1);
    InoperableFlows(q1)=Reduction;

    [ActivitiesCapacity,~,ActorActivities,TotalNrOfActivities,~,~] =
SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows);
    ActivitiesAfterFlowInf(q1,:)=transpose(ActivitiesCapacity); % sparar effekten av en
utslagning av en aktör. Visar statusen för varje aktivitet när aktören (RadNR) är påverkad

```

```

end

ActivitiesDepOnFlows=transpose((1-ActivitiesAfterFlowInf)); % Gör om kapacitet till
nedsättning.
%% Gör om Flöde-aktivitet matris till Flöde-aktör matris
% Aktiviterna blir till aktörer beroende på hur "viktiga" de är för aktören.
FlowDepOnActor=zeros(length(FlowType),LenActors);
for a1=1:length(FlowType)
    for a2=1:LenActors
        for a3=1:TotalNrOfActivities
            if a2==ActorActivities(a3)

FlowDepOnActor(a1,a2)=FlowDepOnActor(a1,a2)+ActivitiesDepOnFlows(a3,a1)*ActivitiesGoalWeight(a
3); % Kolumn utslagen och rad visar hur starkt beroendet är,
            end
        end
    end
end
end
%% Tas bort, bara för att kolla så det räknar rätt
% FlowDepOnActor2=zeros(LenActors,length(FlowType));
% for a1=1:LenActors
%     [~,Col]=find(ActorActivities == a1);
%     for a2=1:length(Col)
%         for a3=1:length(FlowType)
%
FlowDepOnActor2(a1,a3)=FlowDepOnActor2(a1,a3)+ActivitiesDepOnFlows(Col(a2),a3)*ActivitiesGoalW
eight(Col(a2));
%
%     end
% end
%% Skapar vektor som visar flödenas viktighet för kommunen.
ImportanceCityFlows=1:length(FlowType);
for a1=1:length(FlowType)
    ImportanceCityFlows(2,a1)=sum(FlowDepOnActor(a1,:)); % Viktigheten kan alltså vara, så här
långt, [0-Antalet flöden]
end
ImportanceCityFlows=transpose(ImportanceCityFlows); %För att sortera, går säkert ändå men
lättare att förstå :)
[~, idx] = sort(ImportanceCityFlows(:,2)); %Tar reda på hur det ska sorteras.
ImportanceCityRankedFlow = flipud(ImportanceCityFlows(idx,:));
ImportanceCityRankedFlow = ImportanceCityRankedFlow(1:5,:); % Tar fram 5 viktigaste
ImportanceCityRankedFlow(:,2)=int64(100.*ImportanceCityRankedFlow(:,2)/length(FlowType)); %
För att visa faktor (0-100) och i heltal
ImportanceCityRankedFlow=num2cell(ImportanceCityRankedFlow);
for a=1:5 % Namn på 5 viktigaste
    ImportanceCityRankedFlow{a,1}=FlowType{ImportanceCityRankedFlow{a,1}};
end
%% Skapar tabell och text
%f = figure;
colnames = {'Flöde','Beroendefaktor [0-100]'};
t = uitable(f,'Columnwidth',{130 140}, 'Data', ImportanceCityRankedFlow, 'ColumnName',
colnames, ...
'units','normalized', 'pos',[XStartLow YStartLow XStep YStep]);
uicontrol('style','text','units','normalized', 'pos',[XStartLow YStartLow+YStep XStep
0.1], 'stri',{'Kommunens beroende av flöden'}, 'fonts',10 );

%% Skapar vektor som visar flödenas viktighet för specifik aktör. Samma kod som i föregående
avsnitt
if ~isempty(IntrestingActor)
    %for a1=1:length(IntrestingActor)
    FlowImportanceToActor=1:length(FlowType);
    FlowImportanceToActor(2,:)=FlowDepOnActor(:,IntrestingActor);
    %end
    FlowImportanceToActor=transpose(FlowImportanceToActor); %För att sortera, går kanske ändå
men lättare att förstå :)
    FlowImportanceToActor(IntrestingActor,:)=[]; % Tar bort den egna påverkan
    [~, idx] = sort(FlowImportanceToActor(:,2)); %Tar reda på hur det ska sorteras.
    Nedanstående i identiskt med kommunrankningen.
    FlowImportanceToActorRanked = flipud(FlowImportanceToActor(idx,:));
    FlowImportanceToActorRanked = FlowImportanceToActorRanked(1:5,:);
    FlowImportanceToActorRanked(:,2)=int64(100.*FlowImportanceToActorRanked(:,2));
    FlowImportanceToActorRanked=num2cell(FlowImportanceToActorRanked);

    for a=1:5
        FlowImportanceToActorRanked{a,1}=FlowType{FlowImportanceToActorRanked{a,1}};
    end
    % Skapar tabell och text
    t2 = uitable(f,'columnwidth',{130 140}, 'Data', FlowImportanceToActorRanked, 'ColumnName',
colnames, ...
'units','normalized', ...
'pos',[XStartLow+XStep+0.05 YStartLow XStep YStep]);
    uicontrol('style','text','units','normalized', 'pos',[XStartLow+XStep+0.05 YStartLow+YStep
XStep 0.1], 'stri',{Actors{IntrestingActor}.Name}, ' beroende av flöden', 'fonts',10 );
end
end

```

Appendix 1.9 Figur6_creator.m

```

function [ ] = Figur6_Creator( FlowType,Actors)
%Funktionen slår ut en aktör och presenterar hur resterande aktörer är
%koppalde till denna. Med hjälp av funktionen SysIterations presenteras
%även aktörernas kapacitet efter utslagning.
%%TODO
% * Fixa dynamiskt stopp när beroende av aktörer letas, (inte nödvändigt,
% kommer i dagsläget spara max 0.5 sec, 100*antal aktörer sökning nu)
%% vilka tidssteg?
Time=[];
if isempty(Time)
    Time = input('Välj vilken tidsperiod som undersöks:');
end
TimeCategories={'0-2h';'2-8h';'8-24h';'dygn';'veckor';'Månader';'År'};
%% Slå ut:
% Manuellt utslagning. 0 är helt funktionsdugligt. Rad 1 är ID nr, rad 2
% kapacitet. tex [1;0.5] är flöde 1 med 0.5 utslagning
InoperableFlowsNrAndStatus = [0];
InoperableActorsNrAndStatus = [];
%% Slår ut aktörerna och returnerar alla flöden/aktörer och deras kapaciteter
Function=3; % Väljer hur Elimination ska användas
[ InoperableActivities,InoperableFlows,InoperableActorsNrAndStatus,~ ] = Elimination(
InoperableFlowsNrAndStatus, InoperableActorsNrAndStatus,FlowType,Actors,Function);

%% Hämtar variabler som behövs för att bygga beroendematrisen
LenActor=length(Actors);
% Skapar en matris som kopplar ihop aktör-aktör via deras påverkan och
% beroende av flöden
ActorsDepMatrix=zeros(LenActor,LenActor);
for a1=1:LenActor
    for k=1:length(Actors{a1}.DepFlowType)
        if sum(Actors{a1}.DepFlowMatrix{k}(:,Time))>0 % kollar så att beroendet finns i den
undersöka tidsperioden
            DepType=Actors{a1}.DepFlowType(k);
            for a2=1:LenActor
                places=find(DepType==Actors{a2}.InfFlowType,1); %kollar om Aktör a1 är
beroende av ett flöde som Aktör a2 influerar
                if ~isempty(places)
                    if sum(sum(Actors{a2}.InfFlowMatrix{(places)}))>0 % kollar så att
beroendet finns i den undersöka tidsperioden
                        ActorsDepMatrix(a1,a2)=1;
                    end
                end
            end
        end
    end
end
end
end
end
end
%% Spridningskedja
EliminatedActor=InoperableActorsNrAndStatus(1,1);
EliminatedActorTemp=EliminatedActor; %Behövs två för den ena skrivs över
X=0;
Y=0;
ActorPos=cell(LenActor,1);
i = 1; % vilket steg i beroendekedjan som undersöks
NewActors = find(ActorsDepMatrix(:,EliminatedActor)); %vilka aktörer som påverkas av den
utslagna aktören
ActorLine = [];
for k = 1:length(NewActors) % Ger alla beroende aktörer i den här iterationen ID kopplade till
iterationen och påverkande aktör
    ActorLine(k,:) = [i EliminatedActor NewActors(k)]; %
end
if isempty(NewActors) %Avslutar körning om beroende av aktören saknas.
    figure;
    uicontrol('style','text','units','normalized', 'pos',[.0 0.0 1 1],'stri',{'Ingen aktör är
beroende av', Actors{EliminatedActor}.Name, ' under tidsperioden
',char(TimeCategories(Time)),},'fonts',10 );
    return
end

% while-satsen nedan utgår ifrån den utslagna aktören och lagrar vilka som är
% påverkade denna och vilket iteration detta sker. Därefter görs samma
% sak med alla aktörer som finns med i actorline, för alla iterationer.
while i <= LenActor*100 %kör funktionen tills det definitivt är nog, tanken var att det skulle
styras av vilkor längre ner
    EliminatedActorTemp =unique(ActorLine(ActorLine(:,1)==i,3)); % Tar fram aktörer som
påverkas den här iterationen
    EliminatedActorTemp = setdiff(EliminatedActorTemp,ActorLine(:,2)); % Tar bort
EliminatedActorTemp (så att den inte påverkar sig själv)
    i=i+1;
    for j = 1:length(EliminatedActorTemp)
        j = EliminatedActorTemp(j);
        NewActors = find(ActorsDepMatrix(:,j));
        for k = 1:length(NewActors)
            ActorLine(end+1,:) = [i j NewActors(k)]; % [iteration påverkade påverkad]
        end
    end
end
end

```

```

end

ActorLine2=[ActorLine(:,1) ActorLine(:,3)]; % Tar bort kolumn 2 (påverkande aktör)
ActorLineUniq=unique(ActorLine2(:,1:2),'Rows'); % tar bort iteration-aktörer dubletter
[~, idx] = unique(ActorLineUniq(:,2));
ActorLineUniq = ActorLineUniq(idx,:); %Sparar bara första gången en aktör påverkats

for a1=1:size(ActorLineUniq,1) %Ger varje aktör en position,
    %Baserat på iteration vid första påverkan(x) och antal under
    %iterationen (y)
    NrCount=find(ActorLineUniq(:,1)==a1); % Antal som påverkas den här iterationen

    Y=length(NrCount)/2;
    Ymax=Y*2;
    Xstep=0.5/length(NrCount);
    if NrCount(NrCount == EliminatedActor) ~= 0 % Ser till att det inte blir offset i y-led
    när EliminatedActor är med
        Y=Y-0.5;
    end
    PosFix=0;
    for a2=1:length(NrCount)
        if ActorLineUniq(NrCount(a2),2)==EliminatedActor %Fixar så att det inte blir en tom
        pos i y-led för EliminatedActor
            PosFix=1;
        end
        Ya2=-Y+0.5+(a2-1-PosFix);
        Xa2=a1+cos(2*Ya2/Ymax); % Läger aktörerna på en både för att påverkan inom samma
        iteration ska synas
        ActorPos{ActorLineUniq(NrCount(a2),2),1}=[Xa2,Ya2];
    end
end
ActorPos{EliminatedActor}=[0,0]; % Ger den från början utslagna aktören startkoordinaterna

%% Skapar start och stopvärde för de lodrätta linjerna som avgränsar varje spridningsordning
[Row,~]=find((ActorLineUniq(:,2)==EliminatedActor));
ActorLineUniq(Row,:)=[];
for a=1:max(ActorLineUniq(:,1))
    %[ActorsIteration,Col]=find(ActorLineUniq(1,:)==a);
    ActorsIteration=ActorLineUniq(ActorLineUniq(:,1)==a,:);
    for a1=transpose(ActorsIteration(:,2))
        IterationX(a1)=ActorPos{a1}(1);
    end
    IterationX;
    IterationXMM(a,1:2)=[a,min(IterationX)];
    IterationXMM(a,3)=max(IterationX);
end
%% skapa linjer mellan aktörer
LinePos=cell(size(ActorLine,1),1);
for a=1:size(ActorLine,1)
    LinePos{a,1}=[ActorPos{ActorLine(a,2)}, ActorPos{ActorLine(a,3)}];
end

%% Plot
[Xmin,Xmax,Ymin,Ymax]=deal(0);
for a1=1:(size(ActorPos,1)) %Skapar min och max-positioner för figuren, finns nog lättare sätt
    if ~isempty(ActorPos{a1})
        if ActorPos{a1}(1)< Xmin && ActorPos{a1}(1) ~= Inf
            Xmin= ActorPos{a1}(1);
        end
        if ActorPos{a1}(1)> Xmax && ActorPos{a1}(1) ~= Inf
            Xmax = ActorPos{a1}(1);
        end
        if ActorPos{a1}(2)< Ymin && ActorPos{a1}(2) ~= Inf
            Ymin= ActorPos{a1}(2);
        end
        if ActorPos{a1}(2)> Ymax && ActorPos{a1}(2) ~= Inf
            Ymax= ActorPos{a1}(2);
        end
    end
end
end

F=figure;
set(F,'units','normalized','position',[0 .2 .7 .7]) % Gör figuren lite större så att texten
inte överlappar
subplot('position',[0.01 0.01 0.87 0.88]) % GaffeIn
hold on
box on
axis square
grid off
%set(gcf,'Position',[100 80 720 720]);
axis([Xmin+0.57*Ymin Xmax-0.35*Ymin Ymin+0.18*Ymin Ymax-0.18*Ymin])
%set(gca,'xtick',[],'ytick',[]);
Title_string = ['Aktörer som påverkas vid utslagning av ', Actors{EliminatedActor}.Name, ' för
tidsperioden ', lower(char(TimeCategories(Time))), '.', char(10), 'kapacitetsvärden baserat på
fullständig spridning'] ;
title(Title_string)

```

```

for a=1:size(LinePos,1) % Drar pilar mellan beroende aktörer
    if LinePos{a}(1) ~=LinePos{a}(3) || LinePos{a}(2) ~=LinePos{a}(4) % visar inte när en
aktör är beroende av sig själv
        arrow(LinePos{a}(1:2),LinePos{a}(3:4), 'edgecolor',[0.5 0.5 0.5], 'facecolor',[0.5 0.5
0.5], 'Length', 20, 'TipAngle',8)
    end
end
[ActivitiesCapacity,~,ActorActivities,TotalNrOfActivities,~,~] =
SystemIteration(Time,FlowType,Actors,InoperableActivities,InoperableFlows);
ActivitiesDepOnActors=1-ActivitiesCapacity; % gör om kapacitet till nedsättning

%Skapar en vektor med hur många aktiviteter varje aktör har
TotGoalsum=zeros(1,length(Actors));
for a1=1:length(Actors)
    TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
end
% Skapar en vektor med hur "viktigt" varje aktivitet är för aktörens
% måluppfyllnad
ActivitiesGoal=zeros(1,TotalNrOfActivities);
for a1=1:length(Actors)
    for a2=1:length(Actors{a1}.Activities)
        [~,ActorCol]=find(ActorActivities == a1);
        for a3=1:length(ActorCol)
            ActivitiesGoal(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1);
        end
    end
end
end
%% Gör om aktör-aktivitet till aktör-aktör
ActorDepOnActorFinal=zeros(1,length(Actors));
for a1=1:length(Actors)
    for a3=1:TotalNrOfActivities
        if a1==ActorActivities(a3)
            ActorDepOnActorFinal(a1)=ActorDepOnActorFinal(a1)+ActivitiesDepOnActors(a3)*ActivitiesGoal(a3)
        end
    end
end
end
%% Ritar ut lodräta linjer och skriver ut spridningsordningen
set(gca, 'xtick', [], 'ytick', []);
text(0,Ymin+0.18*Ymin-0.14, '0
Ordningen', 'HorizontalAlignment', 'center', 'VerticalAlignment', 'bottom')
for a=1:length(IterationXMM(:,3))
    line([0.8*IterationXMM(a,3),0.8*IterationXMM(a,3)], [Ymin+0.18*Ymin Ymax-
0.18*Ymin], 'color', [0.7 0.7 0.7], 'LineStyle', '- -')
    text(0.8*IterationXMM(a,3)+0.05, Ymin+0.18*Ymin-0.14, [num2str(a), '
Ordningen'], 'HorizontalAlignment', 'left', 'VerticalAlignment', 'bottom')
end
end
%% Plottar allt övrigt
ActorDepOnActorFinal=round(ActorDepOnActorFinal*10)/10;
for a1=1:length(ActorPos) %Sätter ut aktörer och deras namn. (Lite ondigt lång eftersom även
"tomma" aktörer räknas)
    %for a1=sum(~cellfun('isempty',ActorPos))
    Color=RG(ActorDepOnActorFinal(a1)); %Färgsätter röd-gul-grön
    if a1 == EliminatedActor,
        plot(ActorPos{a1}(1),ActorPos{a1}(2), 'square', 'MarkerSize',15, 'MarkerFaceColor',Color, 'Markere
dgeColor', 'black')
        plot(ActorPos{a1}(1),ActorPos{a1}(2), 'x', 'MarkerSize',15, 'MarkerFaceColor', 'black', 'Markeredg
eColor', 'black')
        text(ActorPos{a1}(1),ActorPos{a1}(2)+0.1,Actors{a1}.Name, 'HorizontalAlignment', 'center', 'Verti
calAlignment', 'bottom', 'rotation',0, 'fontweight', 'bold') %Namn
        %text(ActorPos{a1}(1)-0.1,ActorPos{a1}(2)-
0.1,num2str(ActorDepOnActorFinal(a1)), 'HorizontalAlignment', 'Right', 'VerticalAlignment', 'botto
m')
        %% Kapacitet
        elseif ~isempty(ActorPos{a1})
            plot(ActorPos{a1}(1),ActorPos{a1}(2), 'o', 'MarkerSize',12, 'MarkerFaceColor',Color, 'Markeredg
eColor', 'black')
            text(ActorPos{a1}(1),ActorPos{a1}(2)+0.1,Actors{a1}.Name, 'HorizontalAlignment', 'center', 'Verti
calAlignment', 'bottom', 'rotation',0)
            %text(ActorPos{a1}(1)-0.1,ActorPos{a1}(2)-
0.1,num2str(ActorDepOnActorFinal(a1)), 'HorizontalAlignment', 'Right', 'VerticalAlignment', 'botto
m') % Kapacitet
        end
end
end
%% skapa färgförklaring
CustomColor=zeros(100,3);
for a=1:100
    b=a/100;
    CustomColor(a,:)=RG(b);
end
end

```

```
colormap(flipud(CustomColor));
Labels=transpose([0:10:100]);
AxisPos=get(gca,'position');
BarPos=[AxisPos(1)+AxisPos(3)+0.04,AxisPos(2)];
hcb2=colorbar('peer',gca,[0.9 AxisPos(2) 0.015
AxisPos(4)], 'YTickLabel',Labels,'XAxisLocation','bottom','TickLength',[0 0],'Color',[1 0
1]);%,'Position','best');
title(hcb2,['Kapacitet',char(10),'[%]'],'HorizontalAlignment','center');

end
```

Appendix 1.10 FigurBowtie_Creator

```

function [ ] = FigurBowtie_Creator( FlowType,Actors)
%Funktionen slår ut en aktör och presenterar hur resterande aktörer är
%koppalde till denna. Med hjälp av funktionen ActorDepOnActorFinalFunc och SysIterations
presenteras
%även aktörernas kapacitet efter utslagning.
%%ToDo
% * Skulle kunna kalla på figur6 och hämta första positionerna där, men nog
% mer jobb är man sparar
% * Fixa dynamiskt stopp när beroende av aktörer letas, (inte nödvändigt, kommer i dagsläget
spara max 0.5 sec, 100*antal aktörer sökningar nu)
% * Ska pilarna vara snyggare?
if nargin < 2
    FlowType=IndataFlows;
    Actors=IndataActors;
end

%% vilka tidssteg?
Time=[7];
%% Slå ut:
% Manuellt utslagning. 0 är helt funktionsdugligt. Rad 1 är ID nr, rad 2
% kapacitet. tex [1;0.5] är flöde 1 med 0.5 utslagning
InoperableFlowsNrAndStatus = [0];
InoperableActorsNrAndStatus = [];
%% Slår ut aktörerna och returnerar alla flöden/aktörer och deras kapaciteter
Function=2; % Väljer hur Elimination ska användas
[~,~,InoperableActorsNrAndStatus,~] = Elimination( InoperableFlowsNrAndStatus,
InoperableActorsNrAndStatus,FlowType,Actors,Function);

%% Hämtar variabler som behövs för att bygga beroendematrisen
LenActor=length(Actors);
% Skapar en matris som kopplar ihop aktör-aktör via deras påverkan och
% beroende av flöden
k=1:LenActor;
b1 = Actors{1}.InfFlowType;
ActorsDepMatrix=zeros(LenActor,LenActor);
for a1=1:LenActor
    for k=1:length(Actors{a1}.DepFlowType)
        DepType=Actors{a1}.DepFlowType(k);
        for a2=1:LenActor
            places=find(DepType==Actors{a2}.InfFlowType); %kollar om Aktör a1 är beroende av
            ett flöde som Aktör a2 influerar
            if ~isempty(places)
                ActorsDepMatrix(a1,a2)=1;
            end
        end
    end
end
end

%% vänster sida, beroendekedja
EliminatedActor=InoperableActorsNrAndStatus(1,1);
EliminatedActorTemp=EliminatedActor; %Behövs två för den ena skrivs över
X=0;
Y=0;
ActorPos=cell(LenActor,1);
i = 1; % vilket steg i beroendekedjan som undersöks
NewActors = find(ActorsDepMatrix(EliminatedActor,:)); %vilka aktörer som påverkas av den
utslagna aktören
ActorLine = [];
for k = 1:length(NewActors) % Ger alla beroende aktörer i den här iterationen ID kopplade till
iterationen och påverkande aktör
    ActorLine(k,:) = [i EliminatedActor NewActors(k)]; %
end
if isempty(NewActors) %Skriver ut om påverkan av aktören saknas.
    figure;
    uicontrol('style','text','units','normalized','pos',[.0 0.0 1 1],'stri',{'Ingen aktör
påverkar ' , Actors{EliminatedActor}.Name},'fonts',10 );
    disp(['Ingen aktör påverkar ' , Actors{EliminatedActor}.Name])
end
if ~isempty(NewActors) %Avslutar körning om beroende av aktören saknas.
    while i <= LenActor*100 %Kör funktionen tills det definitivt är nog, tanken var att det
skulle styras av vilkor längre ner
        EliminatedActorTemp =unique(ActorLine(ActorLine(:,1)==i,3)); % Tar fram aktörer som
påverkas den här iterationen, ELLER?
        EliminatedActorTemp = setdiff(EliminatedActorTemp,ActorLine(:,2)); % Tar bort
dubbletter av aktörerna ELLER?
        i=i+1;
        for j = 1:length(EliminatedActorTemp)
            j = EliminatedActorTemp(j);
            NewActors = find(ActorsDepMatrix(j,:));
            for k = 1:length(NewActors)
                ActorLine(end+1,:) = [i j NewActors(k)];
            end
        end
    end
end
end

```

```

ActorLine2=[ActorLine(:,1) ActorLine(:,3)]; % Tar bort kolumn 2 (påverkande aktör)
ActorLineUniq=unique(ActorLine2(:,1:2),'Rows'); % tar bort iteration-aktörer dubletter
[~, idx] = unique(ActorLineUniq(:,2));
ActorLineUniq = ActorLineUniq(idx,:); %Sparar bara första gången en aktör påverkats

for a1=1:size(ActorLineUniq,1) %Ger varje aktör en position,
    %Baserat på iteration vid första påverkan(x) och antal under iterationen (y)
    NrCount=find(ActorLineUniq(:,1)==a1); % Antal som påverkas den här iterationen
    Y=length(NrCount)/2;
    Ymax=Y*2;
    Xstep=0.5/length(NrCount);
    if NrCount(NrCount == EliminatedActor) ~= 0 % Ser till att det inte blir offset i y-
led när EliminatedActor är med
        Y=Y-0.5;
    end
    PosFix=0;
    for a2=1:length(NrCount)
        if ActorLineUniq(NrCount(a2),2)==EliminatedActor %Fixar så att det inte blir en
tom pos i y-led för EliminatedActor
            PosFix=1;
        end
        Ya2=-Y+0.5+(a2-1-PosFix);
        Xa2=a1+cos(2*Ya2/Ymax); % Läger aktörerna på en båda för att påverkan inom samma
iteration ska synas
        ActorPos{ActorLineUniq(NrCount(a2),2),1}=[-Xa2, Ya2];
    end
end
ActorPos{EliminatedActor}=[0,0]; % Ger den från början utslagna aktören startkoordinaterna
%% Skapar start och stopvärde för de lodrätta linjerna som avgränsar varje
spridningsordning
[Row,~]=find((ActorLineUniq(:,2)==EliminatedActor));
ActorLineUniq(Row,:)=[];
for a=1:max(ActorLineUniq(:,1))
    ActorsIteration=ActorLineUniq(ActorLineUniq(:,1)==a,:);
    for a1=transpose(ActorsIteration(:,2))
        IterationX(a1)=ActorPos{a1}(1);
    end
    IterationX;
    IterationXMMLeft(a,1:2)=[a,min(IterationX)];
    IterationXMMLeft(a,3)=max(IterationX);
end
%% skapa linjer mellan aktörer
LinePos=cell(size(ActorLine,1),1);
for a=1:size(ActorLine,1)
    LinePos{a,1}=[ActorPos{ActorLine(a,2)}, ActorPos{ActorLine(a,3)}];
end

ActorPosRight=ActorPos;
LinePosRight=LinePos;
end

%% Höger sida, spridningskedja
EliminatedActor=InoperableActorsNrAndStatus(1,1);
EliminatedActorTemp=EliminatedActor; %Behövs två olika för den ena skrivs över
X=0;
Y=0;
ActorPos=cell(LenActor,1);
i = 1; % vilket steg i beroendekedjan som undersöks
NewActors = find(ActorsDepMatrix(:,EliminatedActor)); %vilka aktörer som påverkas av den
utslagna aktören
ActorLine = [];
for k = 1:length(NewActors) % Ger alla beroende aktörer i den här iterationen ID kopplade till
iterationen och påverkande aktör
    ActorLine(k,:) = [i EliminatedActor NewActors(k)]; %
end
if isempty(NewActors) %Avslutar körning om beroende av aktören saknas.
    figure;
    uicontrol('style','text','units','normalized', 'pos',[.0 0.0 1 1],'stri',{'Ingen aktör är
beroende av ', Actors{EliminatedActor}.Name},'fonts',10 );
    disp(['Ingen aktör är beroende av ', Actors{EliminatedActor}.Name])
end
if ~isempty(NewActors)
    while i <= LenActor*100 %Kör funktionen tills det definitivt är nog, tanken var att det
skulle styras av vilkor längre ner
        EliminatedActorTemp =unique(ActorLine(ActorLine(:,1)==i,3)); % Tar fram aktörer som
påverkas den här iterationen
        EliminatedActorTemp = setdiff(EliminatedActorTemp,ActorLine(:,2)); % Tar bort
dubletter av aktörerna
        i=i+1;
        for j = 1:length(EliminatedActorTemp)
            j = EliminatedActorTemp(j);
            NewActors = find(ActorsDepMatrix(:,j));
            for k = 1:length(NewActors)
                ActorLine(end+1,:) = [i j NewActors(k)];
            end
        end
    end
end
end
end

```



```

ActorLine2=[ActorLine(:,1) ActorLine(:,3)]; % Tar bort kolumn 2 (påverkande aktör)
ActorLineUniq=unique(ActorLine2(:,1:2),'rows'); % tar bort iteration-aktörer dubletter
[~, idx] = unique(ActorLineUniq(:,2));
ActorLineUniq = ActorLineUniq(idx,:); %Sparar bara första gången en aktör påverkats

for a1=1:size(ActorLineUniq,1) %Ger varje aktör en position,
    %Baserat på iteration vid första påverkan(x) och antal under iterationen (y)
    NrCount=find(ActorLineUniq(:,1)==a1); % Antal som påverkas den här iterationen
    Y=length(NrCount)/2;
    Ymax=Y*2;
    Xstep=0.5/length(NrCount);
    PosFix=0;
    if NrCount(NrCount == EliminatedActor) ~= 0 % Ser till att det inte blir offset i y-
led när EliminatedActor är med
        Y=Y-0.5;
    end
    for a2=1:length(NrCount)
        if ActorLineUniq(NrCount(a2),2)==EliminatedActor %Fixar så att det inte blir en
tom pos i y-led för EliminatedActor
            PosFix=1;
        end
        Ya2=-Y+0.5+(a2-1-PosFix);
        Xa2=a1+cos(2*Ya2/Ymax); % Läger aktörerna på en båda för att påverkan inom samma
iteration ska synas
        ActorPos{ActorLineUniq(NrCount(a2),2),1}=[Xa2,Ya2];
    end
end
ActorPos{EliminatedActor}=[0,0]; % Ger den från början utslagna aktören startkoordinaterna
%% Skapar start och stopvärde för de lodrätta linjerna som avgränsar varje
spridningsordning
[Row,~]=find((ActorLineUniq(:,2)==EliminatedActor));
ActorLineUniq(Row,:)=[];
for a=1:max(ActorLineUniq(:,1))
    ActorsIteration=ActorLineUniq(ActorLineUniq(:,1)==a,:);
    for a1=transpose(ActorsIteration(:,2))
        IterationX(a1)=ActorPos{a1}(1);
    end
    IterationX;
    IterationXMMRight(a,1:2)=[a,min(IterationX)];
    IterationXMMRight(a,3)=max(IterationX);
end
%% Skapa linjer mellan aktörer
LinePos=cell(size(ActorLine,1),1);
for a=1:size(ActorLine,1)
    LinePos{a,1}=[ActorPos{ActorLine(a,2)}, ActorPos{ActorLine(a,3)}];
end
end

%% Plottar allt övrigt (båda sidor)
%TimeCategories={'0-2h';'2-8h';'8-24h';'Days';'Weeks';'Months';'Year'};
TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'veckor';'Månader';'År'};
[Xmin,Xmax,Ymin,Ymax]=deal(0);
ActorPosMinMax=[ActorPos;ActorPosRight];
ActorPos=ActorPosMinMax;
LinePos=[LinePos;LinePosRight];
for a1=1:(size(ActorPosMinMax,1)) %Skapar min och max-positioner för figuren, finns nog
lättare sätt
    if ~isempty(ActorPosMinMax{a1})
        if ActorPosMinMax{a1}(1)< Xmin && ActorPosMinMax{a1}(1) ~= Inf
            Xmin= ActorPosMinMax{a1}(1);
        end
        if ActorPosMinMax{a1}(1)> Xmax && ActorPosMinMax{a1}(1) ~= Inf
            Xmax = ActorPosMinMax{a1}(1);
        end
        if ActorPosMinMax{a1}(2)< Ymin && ActorPosMinMax{a1}(2) ~= Inf
            Ymin= ActorPosMinMax{a1}(2);
        end
        if ActorPosMinMax{a1}(2)> Ymax && ActorPosMinMax{a1}(2) ~= Inf
            Ymax= ActorPosMinMax{a1}(2);
        end
    end
end

F=figure;
set(F,'units','normalized','outerposition',[0 0 1 1])% Gör figuren lite större så att texten
inte överlappar
subplot
hold on
box on
axis([Xmin+0.3*Ymin Xmax-0.35*Ymin Ymin+0.16*Ymin Ymax-0.16*Ymin])
set(gca,'xtick',[],'ytick',[]);
%% Ritar ut lodrätta linjer och skriver ut spridningsordningen
set(gca,'xtick',[],'ytick',[]);
text(0,Ymin+0.18*Ymin-0.07,'0
Ordningen','HorizontalAlignment','center','VerticalAlignment','bottom')
for a=1:length(IterationXMMRight(:,3)) %Höger sida, påverkan

```

```

    line([0.8*IterationXMMRight(a,3),0.8*IterationXMMRight(a,3)], [Ymin+0.18*Ymin Ymax-
0.18*Ymin], 'color',[0.7 0.7 0.7], 'LineStyle','- -')
    text(0.8*IterationXMMRight(a,3)+0.05, Ymin+0.18*Ymin-0.07, [num2str(a), '
ordningens'], 'HorizontalAlignment', 'left', 'VerticalAlignment', 'bottom')
end
text(mean(IterationXMMRight(:,3)), Ymin+0.18*Ymin-0.14, ['Beroende av
', Actors{EliminatedActor}.Name], 'HorizontalAlignment', 'right', 'VerticalAlignment', 'bottom', 'fo
ntweight', 'bold')
text(mean(IterationXMMLeft(:,2)), Ymin+0.18*Ymin-0.14, ['Påverkan på
', Actors{EliminatedActor}.Name], 'HorizontalAlignment', 'left', 'VerticalAlignment', 'bottom', 'fon
tweight', 'bold')

for a=1:length(IterationXMMLeft(:,2)) %vänster sida, beroende
    line([0.8*IterationXMMLeft(a,2),0.8*IterationXMMLeft(a,2)], [Ymin+0.18*Ymin Ymax-
0.18*Ymin], 'color',[0.7 0.7 0.7], 'LineStyle','- -')
    text(0.8*IterationXMMLeft(a,2)-0.05, Ymin+0.18*Ymin-0.07, [num2str(a), '
ordningens'], 'HorizontalAlignment', 'right', 'VerticalAlignment', 'bottom')
end
%% Ritar ut pilar och namn etc.
Title_string = ['Aktörer som påverkar eller är beroende av ', Actors{EliminatedActor}.Name, '
i någon av tidsperioderna.'];
title(Title_string)
for a=1:size(LinePos,1) % Drar pilar mellan beroende aktörer. Motsatt riktning mot höger sida
    if LinePos{a}(1) ~=LinePos{a}(3) || LinePos{a}(2) ~=LinePos{a}(4) % visar inte när en
aktör är beroende av sig själv
        arrow(LinePos{a}(1:2), LinePos{a}(3:4), 'edgecolor',[0.5 0.5 0.5], 'facecolor',[0.5 0.5
0.5], 'Length', 20, 'TipAngle', 8, 'Ends', 'start')
    end
end
a1Name=0;
for a1=1:length(ActorPos) %Sätter ut aktörer och deras namn. (Lite ondigt lång eftersom även
"tomma" aktörer räknas)
    a1Name=a1;
    if a1>8
        a1Name=a1Name-8;
    end
    if a1Name == EliminatedActor,

plot(ActorPos{a1}(1), ActorPos{a1}(2), 'square', 'MarkerSize', 15, 'MarkerFaceColor', 'white', 'Marke
redgeColor', 'black')

plot(ActorPos{a1}(1), ActorPos{a1}(2), 'x', 'MarkerSize', 15, 'MarkerFaceColor', 'black', 'Markeredg
eColor', 'black')

text(ActorPos{a1}(1), ActorPos{a1}(2)+0.1, Actors{a1Name}.Name, 'HorizontalAlignment', 'center', 'V
erticalAlignment', 'bottom', 'rotation', 0, 'fontweight', 'bold') %Namn
    elseif ~isempty(ActorPos{a1})

plot(ActorPos{a1}(1), ActorPos{a1}(2), 'o', 'MarkerSize', 12, 'MarkerFaceColor', 'white', 'Markeredg
eColor', 'black')

text(ActorPos{a1}(1), ActorPos{a1}(2)+0.1, Actors{a1Name}.Name, 'HorizontalAlignment', 'center', 'V
erticalAlignment', 'bottom', 'rotation', 0)
    end
end
end
end

```

Appendix 1.11 Visualizing_indata.m

```

function [ ] = Visualizing_indata(FlowType, Actors)
%
% Visualiserar indatan, Första delen syftar till att visualisera hur
% aktörens beroende och påverkan förändras över tiden. Andra delen syftar
% till att beskriva hur samtliga aktörer påverkar de olika flödena under
% ett givet tidssteg (eller samtliga). Tredje delen ritar ut vilka aktörer
% som är beroende av andra aktörer. Fjärde delen beskriver vilka flöden som
% är beroende av andra flöden.

% Första, andra och tredje delen är skapad av Jonas och fjärde av Axel och
% Einar

% Todo
% * Gärna kommentar kring hur for:en på rad 296 funkar men i övrigt ser det
% bra ut!
% * Kommentarer: vad betyder "exciterar"? en påverkan
% I fjärde avsnittet göra rätt flöde till flöde..
% Done:
% * Titel på alla figurer
% * Ta bort alla axis-tick
% * Hade infört tomt som input innan så bytte för att vara konsekventa (gamla
% raden fins kvar
% * I tredje avsnittet göra om till arrow pilar (Ser lite queer ut med färgerna men tydligt iaf
% :) Bra riktning på dem med?

%--- LOADING DATA -----
Actor=Actors;
for a=1:length(Actors)
    ActorsNrAndName{a} = [num2str(a) ' : ' char([Actors{a}.Name]) char(10)]; % fullösning.
    Skapar en variabel med alla namn och nummer och printar en per rad
end
disp([char(10),'Välj en aktör att presentera eller lämna tomt för samtliga: '
,char(10),strjoin(ActorsNrAndName)])
%disp([char(10),'Välj ett av följande alternativ: ',char(10),' 0: För samtliga aktörer'
,char(10), strjoin(ActorsNrAndName)])
IntrestingActor = input('Välj vilken aktör som studeras: ');
%t1 = input('Välj tidssteg, för samtliga välj 8:');
t1 = input('Välj tidssteg (1-7), lämna tomt för samtliga:');
%--- FÖRDEFINIERAD INDATA -----
%TimeCategories={'0-2h';'2-8h';'8-24h';'Days';'Weeks';'Months';'Year'};
TimeCategories={'0-2h';'2-8h';'8-24h';'Dygn';'Veckor';'Månader';'Ar'};
%=====
%--- VISUALISERING BEROENDEN -----
%=====
%% --- Visualiserar varje aktörs beroenden och påverkan över tiden
%-----
%if IntrestingActor ==0
if isempty(IntrestingActor)
    for a=1:length(Actor);
        figure('name',(Actor{a}.Name))
        set(gcf,'Position',[250 80 1000 720]);
        m=max(Actor{a}.NbrInfluencingFlows,Actor{a}.NbrDependentFlows);
        color{1}='k'; color{2}='b'; color{3}='r'; color{4}='m'; color{5}='c'; color{6}='y';
        color{7}='g'; color{8}=[0.3 0.3 0.3]; color{9}=[0.6 0.6 0.6]; color{10}=[0.2 0.4 0.6];
        for r=1:Actor{a}.NbrInfluencingFlows
            subplot(m,2,2*r-1)
            title(['Påverkar ' FlowType{Actor{a}.InfFlowType(r)}'],'fontweight','bold')
            hold on
            for t=1:length(Actor{a}.Activities)
                if mod(t,2)==1
                    stairs([1:1:8],[Actor{a}.InfFlowMatrix{r}(t,:)+0.02*rand,'-', 'color',color{t},'Linewidth',2)
                    Actor{a}.InfFlowMatrix{r}(t,end)+0.02*rand,'-', 'color',color{t},'Linewidth',2)
                else
                    stairs([1:1:8],[Actor{a}.InfFlowMatrix{r}(t,:)+0.02*rand,'--', 'color',color{t},'Linewidth',2)
                    Actor{a}.InfFlowMatrix{r}(t,end)+0.02*rand,'--', 'color',color{t},'Linewidth',2)
                end
            end
        end
        axis([1 8 0 1.1])
        set(gca,'xtick',[1.5:1:7.5])
        set(gca,'xtickLabel',TimeCategories)
        ylabel('Störning')
        xlabel('Tidsintervall')
        box on
    end
    color{1}='k'; color{2}='b'; color{3}='r'; color{4}='m'; color{5}='c'; color{6}='y';
    for s=1:Actor{a}.NbrDependentFlows
        subplot(m,2,2*s)
        title(['Beroende av ' FlowType{Actor{a}.DepFlowType(s)}'],'fontweight','bold')
        hold on
        for t=1:length(Actor{a}.Activities)
            if mod(t,2)==1
                stairs([1:1:8],[Actor{a}.DepFlowMatrix{s}(t,:)+0.02*rand,'-', 'color',color{t},'Linewidth',2)
                Actor{a}.DepFlowMatrix{s}(t,end)+0.02*rand,'-', 'color',color{t},'Linewidth',2)
            else
                stairs([1:1:8],[Actor{a}.DepFlowMatrix{s}(t,:)+0.02*rand,'--', 'color',color{t},'Linewidth',2)
                Actor{a}.DepFlowMatrix{s}(t,end)+0.02*rand,'--', 'color',color{t},'Linewidth',2)
            end
        end
    end
end
end

```

```

                stairs([1:1:8],[Actor{a}.DepFlowMatrix{s}(t,:)+0.02*rand,'--','Color',color{t},'Linewidth',2)
Actor{a}.DepFlowMatrix{s}(t,end)]+0.02*rand,'--','Color',color{t},'Linewidth',2)
            end
        end
        axis([1 8 0 1.1]) % Sätter rätt axlar på subplottarna, tidsstegen och till vilken
påverkans grad.
        set(gca,'xtick',[1.5:1:7.5])
        set(gca,'xtickLabel',TimeCategories)
        ylabel('Störning')
        xlabel('Tidsintervall')
        box on
    end
    legend(Actor{a}.Activities)
    set(gca,'xtick',[1.5:1:7.5])
    set(gca,'xtickLabel',TimeCategories)
    ylabel('Störning')
    xlabel('Tidsintervall')
end
else
figure('name',(Actor{IntrestingActor}.Name))
set(gcf,'Position',[250 80 1000 720]);

m=max(Actor{IntrestingActor}.NbrInfluencingFlows,Actor{IntrestingActor}.NbrDependentFlows);
color{1}='k'; color{2}='b'; color{3}='r'; color{4}='m'; color{5}='c'; color{6}='y';
color{7}='g'; color{8}=[0.3 0.3 0.3]; color{9}=[0.6 0.6 0.6]; color{10}=[0.2 0.4 0.6];
for r=1:Actor{IntrestingActor}.NbrInfluencingFlows

    subplot(m,2,2*r-1)
    title(['Påverkans '])
    FlowType{Actor{IntrestingActor}.InfFlowType(r)}, 'fontweight','bold')
    hold on
    for t=1:length(Actor{IntrestingActor}.Activities)
        if mod(t,2)==1
            Actor{IntrestingActor}.InfFlowMatrix{r}(t,:)+0.02*rand,'--','Color',color{t},'Linewidth',2)
        else
            Actor{IntrestingActor}.InfFlowMatrix{r}(t,:)+0.02*rand,'--','Color',color{t},'Linewidth',2)
        end
    end
    axis([1 8 0 1.1]) % Sätter rätt axlar på subplottarna, tidsstegen och till vilken
påverkans grad.
    set(gca,'xtick',[1.5:1:7.5])
    set(gca,'xtickLabel',TimeCategories)
    ylabel('Störning')
    xlabel('Tidsintervall')
    box on
end
color{1}='k'; color{2}='b'; color{3}='r'; color{4}='m'; color{5}='c'; color{6}='y';
for s=1:Actor{IntrestingActor}.NbrDependentFlows
    subplot(m,2,2*s)
    title(['Beroende av '])
    FlowType{Actor{IntrestingActor}.DepFlowType(s)}, 'fontweight','bold')
    hold on
    for t=1:length(Actor{IntrestingActor}.Activities)
        if mod(t,2)==1
            Actor{IntrestingActor}.DepFlowMatrix{s}(t,:)+0.02*rand,'--','Color',color{t},'Linewidth',2)
        else
            Actor{IntrestingActor}.DepFlowMatrix{s}(t,:)+0.02*rand,'--','Color',color{t},'Linewidth',2)
        end
    end
    axis([1 8 0 1.1]) % Sätter rätt axlar på subplottarna, tidsstegen och till vilken
påverkans grad.
    set(gca,'xtick',[1.5:1:7.5])
    set(gca,'xtickLabel',TimeCategories)
    ylabel('Störning')
    xlabel('Tidsintervall')
    box on
end
legend(Actor{IntrestingActor}.Activities)
set(gca,'xtick',[1.5:1:7.5])
set(gca,'xtickLabel',TimeCategories)
St=suptitle(['Flödespåverkan och flödesberoende för ',(Actor{a}.Name)]);
set(St,'FontSize',10); % 'Fontweight','bold');
end
%-----
%% --- Visualiserar aktörernas beroenden till flöden i de olika tidsstegen
%-----
%Initierar värden
for k=1:length(FlowType)
    FlowsY(k)=67+3-k*3;
end
ActorsXY=[...
    25 36 47 58 73 85 95 107 117;
    85 85 85 85 85 85 85 85 85];

```

```

%Går igenom och bokför hur många beroenden det är till respektive flöde för respektie
tidsperiod
%--> En Matris(antal flöden, antal tidsaspekter) där nummer anger antal aktörer som påverkar
flödet
%+ En Matris(antal flöden, antal tidsaspekter) där nummer anger antal aktörer som beror av
flödet
FlowMatrixInf=zeros(length(FlowType),length(TimeCategories));
FlowMatrixDep=zeros(length(FlowType),length(TimeCategories));
if isempty(t1)
    %if t1==8
    t1=1:7;
end
for TimePeriod=t1
    for a=1:length(Actor)
        NbrInfFlows=Actor{a}.NbrInfluencingFlows;
        for r=1:NbrInfFlows
            StrengthFlowInf=max(Actor{a}.InfFlowMatrix{r}(:,TimePeriod));

FlowMatrixInf(Actor{a}.InfFlowType(r),TimePeriod)=FlowMatrixInf(Actor{a}.InfFlowType(r),TimePe
riod)+1*StrengthFlowInf;
            end
            NbrDepFlows=Actor{a}.NbrDependentFlows;
            for r=1:NbrDepFlows
                StrengthFlowDep=max(Actor{a}.DepFlowMatrix{r}(:,TimePeriod));

FlowMatrixDep(Actor{a}.DepFlowType(r),TimePeriod)=FlowMatrixDep(Actor{a}.DepFlowType(r),TimePe
riod)+1*StrengthFlowDep;
            end
        end
    end
    %Plottar en figur för varje tidsperiod
    for TimePeriod=t1
        figure
        title(['Aktörernas beroenden och påverkan på samtliga flöden under tidsperioden '
TimeCategories{TimePeriod},char(10),'Grönt visar beroende och rött påverkan'])
        hold on
        set(gcf,'Position',[100 80 1200 720]);
        set(gca,'xtick',[],'ytick',[]); % Skippa axlarnas texter
        %Plottar flödena
        for k=1:length(FlowType)
            widthLine=0.5+0.5*FlowMatrixDep(k,TimePeriod);
            plot([20 110],[FlowsY(k) FlowsY(k)],'k-', 'Linewidth',widthLine)
            text(2.5,FlowsY(k),FlowType{k}, 'Fontweight','Bold')
        end
        %Plottar aktörer och beroenden
        for a=1:length(Actor)
            NbrInfFlows=Actor{a}.NbrInfluencingFlows;
            %vertikal linje höger ut från noden
            plot([ActorsXY(a,1) ActorsXY(a,1)+1+0.7*NbrInfFlows],[ActorsXY(a,2) ActorsXY(a,2)], 'r-
')
            %vertikal linje ner till flödet
            for r=1:NbrInfFlows
                MagnitudeInfFlow=max(Actor{a}.InfFlowMatrix{r}(:,TimePeriod));
                if MagnitudeInfFlow>0
                    plot([ActorsXY(a,1) ActorsXY(a,1)+1+0.7*r],[ActorsXY(a,2)
FlowsY(Actor{a}.InfFlowType(r))], 'r-', 'Linewidth',3*MagnitudeInfFlow)
            end
            plot([ActorsXY(a,1)+1+0.7*r,[FlowsY(Actor{a}.InfFlowType(r))], 'ok', 'MarkerSize',2, 'MarkerFace
Color','r')
        end
        NbrDepFlows=Actor{a}.NbrDependentFlows;
        %vertikal linje vänster ut från noden
        plot([ActorsXY(a,1)-1-0.7*NbrDepFlows ActorsXY(a,1)],[ActorsXY(a,2) ActorsXY(a,2)], 'g-
')
        %vertikal linje ner till flödet
        for r=1:NbrDepFlows
            MagnitudeDepFlow=max(Actor{a}.DepFlowMatrix{r}(:,TimePeriod));
            if MagnitudeDepFlow>0
                plot([ActorsXY(a,1) ActorsXY(a,1)-1-0.7*r],[ActorsXY(a,2)
FlowsY(Actor{a}.DepFlowType(r))], 'g-', 'Linewidth',3*MagnitudeDepFlow)
                plot([ActorsXY(a,1)-1-
0.7*r],[FlowsY(Actor{a}.DepFlowType(r))], 'ok', 'MarkerSize',2, 'MarkerFaceColor','g')
            end
        end
        plot(ActorsXY(a,1),ActorsXY(a,2), 'ko', 'MarkerSize',12, 'MarkerFaceColor',[0.8 0.8 0.8])
        text(ActorsXY(a,1),ActorsXY(a,2),Actor{a}.Name, 'HorizontalAlignment','center', 'VerticalAligme
nt','middle', 'Rotation',90, 'Fontweight','bold', 'FontSize',11)
    end
    axis([0 112 0 100])
end

%-----
%% ---- Plotta varje aktör och vilka andra aktörer de är beroende av för samtliga tidssteg--
%-----

```

```

%Skapar en dependency matrix(actor,actor) där en etta innebär att actor 1 (rad) är beroende av
actor 2 (kolumn)
angles=[0:2*pi/length(Actor):2*pi];
for k=1:length(Actor)
    ActorsXY(k,:)=40.*[cos(angles(k)) ; sin(angles(k))];
end
ActorDepMatrix=zeros(length(Actor),length(Actor));
for a1=1:length(Actor)
    for k=1:length(Actor{a1}.DepFlowType)
        DepType=Actor{a1}.DepFlowType(k);
        for a2=1:length(Actor)
            places=find(DepType==Actor{a2}.InfFlowType); %Kollar om Aktör a1 är beroende av
            ett flöde som Aktör a2 influerar
            if ~isempty(places)
                ActorDepMatrix(a1,a2)=1;
            end
        end
    end
end
end
%Plotar om en aktör är beroende av ett flöde som en annan aktör påverkar (först ordningens
beroende)
figure;
hold on
box on
axis square
set(gcf,'Position',[100 80 720 720]);
set(gca,'xtick',[],'ytick',[]);
axis([-60 60 -60 60])
ActorColours=jet(length(Actor));
[m,n]=size(ActorDepMatrix);
for a1=1:m
    for a2=1:n
        if a1~=a2
            if ActorDepMatrix(a1,a2)==1
                arrow([ActorsXY(a1,1) ActorsXY(a1,2)], [ActorsXY(a2,1)
ActorsXY(a2,2)], 'edgecolor','black','facecolor',ActorColours(a2,:), 'Length',
30,'TipAngle',8,'Ends','start')
            end
        end
    end
end
end
title(' Direkta kopplingar mellan aktörerna via ett flöde')
for a=1:length(Actor)
    plot(ActorsXY(a,1),ActorsXY(a,2),'ko','MarkerSize',12,'MarkerFaceColor',ActorColours(a,:))
end
text(ActorsXY(a,1),ActorsXY(a,2)+2,Actor{a}.Name,'HorizontalAlignment','center','VerticalAlign
ment','bottom')
end

%% Flöde till flöde - Skapad av Axel och Einar
angles=[0:2*pi/length(FlowType):2*pi];
for k=1:length(FlowType)
    ActorsXY(k,:)=40.*[cos(angles(k)) ; sin(angles(k))];
end
[ActivitesInfFlows, ActivitesDepFlows,~,~]=CreateMatrix(7,FlowType,Actors); % Tidsperiod 7 är
väld för att om det finns ett beroende eller påverkan för någon tidsperiod finns det även för
tidsperiod 7.
[m,n]=size(ActivitesInfFlows);
FlowInfMatrix=zeros(m,m);
ActivitesAnyInfOnFlow=zeros(m,n);
ActivitesAnyDepOnFlow=zeros(m,n);

for a1=1:m
    for a2=1:n
        if ActivitesInfFlows(a1,a2)>0
            ActivitesAnyInfOnFlow(a1,a2)=1; % Om någon aktivitet påverkar ett flöde lite
            exciterar en påverkan
        end
        if ActivitesInfFlows(a1,a2)==0
            ActivitesAnyInfOnFlow(a1,a2)=0.1; % Löser så att det inte skapas en koppling om
            det är noll i båda matriserna.
        end
        if ActivitesDepFlows(a1,a2)>0
            ActivitesAnyDepOnFlow(a1,a2)=1; % Om någon aktivitet är lite beroende av ett
            flöde exciterar ett beroende
        end
    end
end
end

for b1=1:m
    for b2=1:n
        for b3=1:m
            if ActivitesAnyDepOnFlow(b1,b2)==ActivitesAnyInfOnFlow(b3,b2)
                FlowInfMatrix(b1,b3)=1;
            end
        end
    end
end

```

```

    end

end
%Plotar om en aktör är beroende av ett flöde som en annan aktör påverkar (först ordningens
beroende)
figure
hold on
box on
axis square
title('Direkta kopplingar mellan flöden')
set(gcf,'Position',[100 80 720 720]);
set(gca,'xtick',[],'ytick',[]); % tar bort texten på axlarna
axis([-60 60 -60 60])
ActorColours=hsv(length(FlowType));
[m,n]=size(FlowInfMatrix);
for a1=1:m % Ritar linjer
    for a2=1:n
        if a1~=a2
            if FlowInfMatrix(a1,a2)==1
                arrow([ActorsXY(a1,1) ActorsXY(a1,2)], [ActorsXY(a2,1)
ActorsXY(a2,2)], 'edgecolor','black','facecolor',ActorColours(a1,:), 'Length',
30, 'TipAngle',8, 'Ends', 'stop')
            end
        end
    end
end
end

for a=1:m % Ritar ut punkter och namn
    plot(ActorsXY(a,1),ActorsXY(a,2), 'ko', 'MarkerSize',12, 'MarkerFaceColor',ActorColours(a,:))

text(ActorsXY(a,1),ActorsXY(a,2)+2, char(FlowType(a)), 'HorizontalAlignment','center', 'VerticalA
lignment','bottom')
end
end

```

Appendix 1.12 Responsgraf.m

```

function [] = Responsgraf(FlowType,Actors)
%% Skapar figurer som visar resultatet av en/alla aktörer då en aktör blir stegvis utstörd
%Printa=1; Visar alla aktörers nedsättning när en sänks
%Printa=2; Visar en aktörs nedsättning med min-max boxplot

if nargin <2
FlowType=IndataFlows;
disp('Kör utan att hämta indata från huvudfunktionen')
Actors=IndataActors;
end
% Manuella input
Steps=10; % Ändrar hur många steg som ska slås ut. 10 är snabbt, 100 detaljerat och tar ~1 min
och figur 2 blir kass.
%% Hämtar variabler som behövs för att bygga beroendematrisen
Time=[];
if isempty(Time)
Time = input('Välj vilken tidsperiod som undersöks:');
end
[~,~,~,TotalNrOfActivities,~,~] = SystemIteration(Time,FlowType,Actors,0,0);
LenActors=length(Actors);
ActorsDepOnActivities=zeros(LenActors,TotalNrOfActivities); % Skapar en matris som används
senare
InoperableFlowsNrAndStatus=[0];
%% Input
disp(['...
'1. Visar alla aktörers nedsättning när en aktör sänks stegvis',char(10),...
'2. Visar nedsättningen hos en aktör stegvis inklusive minsta och högsta värden',char])
Printa=input('Välj output:');
%Time=input('Välj tidsperiod:');
for a=1:LenActors
ActorsNrAndName{a} = [ num2str(a) ' : ' char(Actors{a}.Name) char(10)]; % variabel med alla
namn och nummer och printar en per rad
end
disp(['Möjliga aktörer:' ,char(10), strjoin(ActorsNrAndName)])
DisturbedActorA1=input('Aktör att slå ut (Temporärt 1 är bra:');
if Printa==2
CurrentActor=input('Välj aktör att presentera: (ta 4):');
end
%%
for MM=1:3 % För att välja MaxmedelMin output från systemiterationer
switch MM
case 1
MinMax='Min';
case 2
MinMax='Mean';
case 3
MinMax='Max';
end
ActorsState=[];
%A1=1; % utslagen aktör

for A2=1:Steps
InoperableActorsNrAndStatus=[DisturbedActorA1;A2/Steps]; %Aktör och sänkning
[ InoperableActivities,InoperableFlows,~, InoperableFlowsNrAndStatus ] = Elimination(
InoperableFlowsNrAndStatus, InoperableActorsNrAndStatus,FlowType,Actors);
[ActivitiesCapacity,~,ActorActivities,TotalNrOfActivities,~,~] =
SystemIteration(Time,FlowType,Actors,InoperableActivities,InoperableFlows,MinMax);
ActorsDepOnActivities(DisturbedActorA1,:)=transpose(ActivitiesCapacity(:)); % Sparar
effekten av en utslagning av en aktör. Visar statusen för varje aktivitet när aktören (RadNR)
är påverkad
ActivitiesDepOnActors=transpose((1-ActorsDepOnActivities)); %Gör om kapacitet till
nedsättning.
% Visar alla aktiviteters nedsättningar när aktörerna(kolumnerna) är
%Skapar en vektor med hur många målpoäng varje aktör har
TotGoalsum=zeros(1,LenActors);
for a1=1:LenActors
TotGoalsum(a1)=sum(sum(Actors{a1}.MatrixActGoals));
end
%Skapar en vektor med hur "viktig" varje aktivitet är för aktörens måluppyllnad
ActivitiesGoalWeight=zeros(1,TotalNrOfActivities);
for a1=1:LenActors
for a2=1:length(Actors{a1}.Activities)
[~,ActorCol]=find(ActorActivities == a1);
for a3=1:length(ActorCol)
ActivitiesGoalWeight(ActorCol(a3))=sum(Actors{a1}.MatrixActGoals(a3,:))/TotGoalsum(a1);
end
end
end
%% Gör om aktör-aktivitet matris till aktör-aktör matris
% Aktiviteterna blir till aktörer beroende på hur "viktiga" de är för aktören.
ActorDepOnActor=zeros(LenActors,LenActors);
for a1=1:LenActors
for a2=1:LenActors
for a3=1:TotalNrOfActivities

```



```

        if a1==ActorActivities(a3)
ActorDepOnActor(a1,a2)=ActorDepOnActor(a1,a2)+ActivitiesDepOnActors(a3,a2)*ActivitiesGoalWeight(a3);
        end
    end
end
    ActorsState(:,A2)=ActorDepOnActor(:,DisturbedActorA1);
end
ActorsStateMinMax(:, :, MM)=ActorsState;
end
for a=1:LenActors
    ActorsNames{a} = [char(Actors{a}.Name)]; % Variabel med alla namn och nummer och printar en per rad
end
%% Väljer figur att printa
%Printa=1; Visar alla aktörers nedsättning när en sänks
%Printa=2; Visar en aktörs nedsättning med min-max boxplot
figure('units','normalized','Position',[0.1 0.1 0.6 0.6]);

hold on
X=(1/Steps:1/Steps:1);
ActorColours=jet(length(Actors));
switch Printa
    case 1
        a=2; % för att bara visa medelvärden. for a=1:3 % visar min/mean/max
        for a1=1:length(Actors)
            plot(X,squeeze(ActorsStateMinMax(a1,:,a)), 'color', ActorColours(a1,:))
        end
        title(['Samtliga aktörers nedsättning vid stegvis utslagning av ',ActorsNames{DisturbedActorA1} ])
        axis('square')
        set(gca,'Xtick',[0:0.1:1])
        box on
        xlim([1/Steps 1])
        ylim([1/Steps 1])
        xlabel(['Störning av ',ActorsNames{DisturbedActorA1}])
        ylabel('Störning som respons hos övriga aktörer')
        set(gca,'Ytick',[0:0.1:1])
        %plot(X,ActorsState(:,1), 'color', 'black')
        ActorsNames{DisturbedActorA1}=[ActorsNames{DisturbedActorA1}, ' (Direkt påverkad)'];
        %set(hFig,'units','normalized','Position',[0.1 0.1 0.6 0.6])
        legend(ActorsNames,'Location','BestOutside')
        %end
    case 2
        PlotMatrix=flipud(transpose(squeeze(ActorsStateMinMax(CurrentActor, :, :))));
        boxplot(PlotMatrix,X)
        plot(ActorsStateMinMax(CurrentActor, :, 2));
        title(['Respons hos ',ActorsNames{CurrentActor} , ' vid stegvis utslagning av ',ActorsNames{DisturbedActorA1}, ' ', char(10), 'Lådorna visar 25 %, 50 % och 75 % av spridningen på respons när högsta respektive lägsta indatavärde används för beroende och påverkan.' ])
        axis('square')

        %ylim([0.5 1.1])
        xlabel(['Störning av ',ActorsNames{DisturbedActorA1}])
        ylabel(['Störning som respons hos ',ActorsNames{CurrentActor}])

        %set(gca,'xLabel',['Nedsättning hos hos',ActorsNames{A1}] )
        set(gca,'Ytick',[0:0.1:1])
end
end

```

Appendix 1.13 Elimination.m

```

function [
InoperableActivities,InoperableFlows,InoperableActorsNrAndStatus,InoperableFlowsNrAndStatus ]
= Elimination( InoperableFlowsNrAndStatus,
InoperableActorsNrAndStatus,FlowType,Actors,Function)
%UNTITLED5 Summary of this function goes here
% Funtionen frågar efter input kring utslagna flöden och aktörer och
% returnerar vilka en vektor med påverkade flöden och en vektor där
% påverkade aktör översatts till påverkad aktivitet. Funktionen kan köras
% i tre olika lägen för aktörsinmatning (Styrs med Function):
%     1. Frågar efter flera nedsatta
%     2. Frågar efter en nedsatt och sätter störningen 1. (För Bowtie)
%     3. Frågar efter en nedsatt och dess störning
%
%%
[~,~,ActorActivities,TotalNrOfActivities]=CreateMatrix(1,FlowType,Actors); % Används bara för
att generera raden över utslagning
LenFlow= length(FlowType);
InoperableFlows=zeros(LenFlow,1);
InoperableActivities= zeros(1,TotalNrOfActivities);
if nargin < 5
    Function=1; % Används för att Elimination ska kunna användas och bara fråga efter en aktör
och inte utslagningen
end
%% Utslagning av flöden. Via input
if InoperableFlowsNrAndStatus == 0 % bara för att det ska gå att köra utan utslagna
flöde/aktörer

elseif isempty(InoperableFlowsNrAndStatus)
    for a3=1:LenFlow
        FlowsNrAndName{a3} = [ num2str(a3) ':' char(FlowType(a3)) char(10)]; % Skapar en
variabel med alla namn och nummer och printar en per rad
    end

    disp(['Alla flöden:' ,char(10), strjoin(FlowsNrAndName)]);
    InoperableFlowsNrAndStatus = input('Flöde att slå ut (fyll i [Nr1 Nr2]):');
    if InoperableFlowsNrAndStatus ~= 0

        % InoperableFlowsNr: rad 1 är numret på den påverka aktören och rad 2 är
% storleken på utslagningen
        for a3=1:size(InoperableFlowsNrAndStatus,2)
            InoperableFlowsNrAndStatus(2,a3) =
input([char(FlowType(InoperableFlowsNrAndStatus(1,a3))) ' nedsättning: ']); % eller störning

        end
    end
end
% Sätter nedsättning på alla flöden som är påverkade
if InoperableFlowsNrAndStatus ~= 0
    for a3=1:size(InoperableFlowsNrAndStatus,2)
        InoperableFlows(InoperableFlowsNrAndStatus(1,a3))=InoperableFlowsNrAndStatus(2,a3);
    end
end

%% Aktörer. Utslagning via input
if InoperableActorsNrAndStatus == 0
elseif isempty(InoperableActorsNrAndStatus)
    for a3=1:length(Actors)
        ActorsNrAndName{a3} = [ num2str(a3) ':' char([Actors{a3}.Name]) char(10)]; %
fullösning. Skapar en variabel med alla namn och nummer och printar en per rad
    end
    disp(['Alla aktörer:' ,char(10), strjoin(ActorsNrAndName)]);
    if Function == 1;
        InoperableActorsNrAndStatus = input('Aktör att slå ut (fyll i [Nr1 Nr2]):');
        if InoperableActorsNrAndStatus ~= 0
            for a3=1:size(InoperableActorsNrAndStatus,2)
                InoperableActorsNrAndStatus(2,a3) =
input([char(Actors{InoperableActorsNrAndStatus(1,a3)}.Name) ' nedsättning: ']);

            end
        end
    end
    if Function > 1;
        InoperableActorsNrAndStatus = input('Välj aktör att utgå ifrån:');
        InoperableActorsNrAndStatus(2,1)=1;
        if Function ==3
            InoperableActorsNrAndStatus(2,1) =
input([char(Actors{InoperableActorsNrAndStatus(1,1)}.Name) ' nedsättning: ']);
        end
    end
end
end
end

```

```
if InoperableActorsNrAndStatus ~= 0
    for a3=1:size(InoperableActorsNrAndStatus,2)
        InoperableActors(InoperableActorsNrAndStatus(1,a3))=InoperableActorsNrAndStatus(2,a3);
    end
    % Kolla vilka aktörer, InoperableActors, som är utslagna och applicera utslagningen på
    alla
    % aktiviteterna, InoperableActivities, med hjälp av ActorActivities,
    for a3=1:length(InoperableActors)
        if InoperableActors(a3) > 0
            %ta reda på alla positioner i ActorActivities som innehåller a, dvs
            %numret på den utslagna aktören
            [~,ActorCol]=find(InoperableActorsNrAndStatus(1,:) == a3);
            [ActivRow,ActiveCol]=find(ActorActivities == a3);
            InoperableActivities(ActivRow,ActiveCol)=InoperableActorsNrAndStatus(2,ActorCol);
        % här ska kapaciteten efter nedsättning fyllas i.
        end
    end
end

end
```

Appendix 1.14 CreateMatrix.m

```

%% To do
% 2. Att använda find istället för for och if på rad 46 och 52 hade antagligen snabbat upp
function [ ActivitesInfFlows, ActivitesDepFlows, ActorsActivites, TotalNrOfActivites ] =
CreateMatrix(t,Flows,Actors,MinMax)
% Funktionen skapar matrisen med alla aktiviteters påverka och beroende
% av flödena (flöde x aktiviteter), vilka aktiviteter som tillhör vilken
% aktör (vektor med aktörens IDnr på varje position som är aktörens
% aktivitet i de andra matriserna och det totala antalet aktiviteter
FlowType=Flows;
Actor=Actors;
LenFlow= length(FlowType);
LenActor= length(Actor);
if nargin < 4 % För att låta input av MinMax vara frivilligt (ingen input ger mean tillbaka)
    MinMax=[];
end

%% Skapar en vektor som visar aktörtillhörighet för aktiviteter i ActivitesInfFlows och
ActivitesDepFlows.
% Skapar även TotalNrOfActivites
TotalNrOfActivites = 0;
ActorsActivites=[];
Stop=1;
for a1=1:length(Actor)
    Start = TotalNrOfActivites+1;
    TotalNrOfActivites = TotalNrOfActivites + length(Actor{a1}.Activities);
    Stop = length(ActorsActivites)+ length(Actor{a1}.Activities);
    ActorsActivites(Start:Stop)= a1;
end

%% Skapar matriserna ActivitesInfFlows och ActivitesDepFlows
ActivitesInfFlows=zeros(LenFlow,TotalNrOfActivites);
ActivitesDepFlows=zeros(LenFlow,TotalNrOfActivites);
% Hämta data från varje (Actor{a}.InfFlowMatrix{b} för att sammanställa i
% ActivitesInfFlows, hur aktiviteter påverkar flödena Och en vektor/matrix
% som berättar vilka kolumner i matrisen som tillhör vilken aktör. Vektorn
% är lika långt som totala antalet aktiviter och innehålla en aktörs nummer
% på varje aktivits plats.
for a1=1:LenFlow
    Stop=0;% Nollställa Stop för varje flöde.
    for a2=1:LenActor
        for a3=1:(Actor{a2}.NbrInfluencingFlows)
            if Actor{a2}.InfFlowType(a3) == a1
                Start = length(ActorsActivites(ActorsActivites < a2))+1;
                Stop = Start-1+length(transpose((Actor{a2}.InfFlowMatrix{a3}(:,t))));
                ActivitesInfFlows(a1,Start:Stop) =
                    transpose((Actor{a2}.InfFlowMatrix{a3}(:,t)));
            end
        end
    end
end

% Gör ovanstående för beroendematrisen
for a1=1:LenFlow
    Stop=0;% ny måste nollställa Stop för varje flöde.
    for a2=1:LenActor
        for a3=1:(Actor{a2}.NbrDependentFlows)
            if Actor{a2}.DepFlowType(a3) == a1
                Start = length(ActorsActivites(ActorsActivites < a2))+1;
                Stop = Start-1+length(transpose((Actor{a2}.DepFlowMatrix{a3}(:,t))));
                ActivitesDepFlows(a1,Start:Stop) =
                    transpose((Actor{a2}.DepFlowMatrix{a3}(:,t)));
            end
        end
    end
end

%% För att returnera Min och Max om detta efterfrågats
if strcmpi(MinMax,'Min') % Ej case sensitive
    [ ActivitesDepFlows, ~, ActivitesInfFlows, ~ ] = MinMaxCreator( ActivitesDepFlows,
ActivitesInfFlows );
elseif strcmpi(MinMax,'Max')
    [ ~, ActivitesDepFlows, ~, ActivitesInfFlows ] = MinMaxCreator( ActivitesDepFlows,
ActivitesInfFlows );
end
end

```

Appendix 1.15 MinMaxCreator.m

```

function [ MinActivitesDepFlows, MaxActivitesDepFlows, MinActivitesInfFlows,
MaxActivitesInfFlows ] = MinMaxCreator( ActivitesDepFlows, ActivitesInfFlows )
%MinMaxCreator skapar min och max värde för utgångsmodellerna
%Genom att forloopa ActivitesDepFlows och ActivitesInfFlows matriserna
%bytts siffrorna ut till lägsta respektive högsta värdet

[b1,b2]=(size(ActivitesDepFlows));
for a1=1:b1
    for a2=1:b2
        if ActivitesDepFlows(a1,a2) == 0
            MinActivitesDepFlows(a1,a2)=0;
            MaxActivitesDepFlows(a1,a2)=0;
        end
        if ActivitesInfFlows(a1,a2) == 0
            MinActivitesInfFlows(a1,a2)=0;
            MaxActivitesInfFlows(a1,a2)=0;
        end
        if ActivitesDepFlows(a1,a2) == 0.13
            MinActivitesDepFlows(a1,a2)=0.01;
            MaxActivitesDepFlows(a1,a2)=0.25;
        end
        if ActivitesInfFlows(a1,a2) == 0.13
            MinActivitesInfFlows(a1,a2)=0.01;
            MaxActivitesInfFlows(a1,a2)=0.25;
        end
        if ActivitesDepFlows(a1,a2) == 0.38
            MinActivitesDepFlows(a1,a2)=0.26;
            MaxActivitesDepFlows(a1,a2)=0.50;
        end
        if ActivitesInfFlows(a1,a2) == 0.38
            MinActivitesInfFlows(a1,a2)=0.26;
            MaxActivitesInfFlows(a1,a2)=0.50;
        end
        if ActivitesDepFlows(a1,a2) == 0.63
            MinActivitesDepFlows(a1,a2)=0.51;
            MaxActivitesDepFlows(a1,a2)=0.75;
        end
        if ActivitesInfFlows(a1,a2) == 0.63
            MinActivitesInfFlows(a1,a2)=0.51;
            MaxActivitesInfFlows(a1,a2)=0.75;
        end
        if ActivitesDepFlows(a1,a2) == 0.88
            MinActivitesDepFlows(a1,a2)=0.76;
            MaxActivitesDepFlows(a1,a2)=0.99;
        end
        if ActivitesInfFlows(a1,a2) == 0.88
            MinActivitesInfFlows(a1,a2)=0.76;
            MaxActivitesInfFlows(a1,a2)=0.99;
        end
        if ActivitesDepFlows(a1,a2) == 1
            MinActivitesDepFlows(a1,a2)=1;
            MaxActivitesDepFlows(a1,a2)=1;
        end
        if ActivitesInfFlows(a1,a2) == 1
            MinActivitesInfFlows(a1,a2)=1;
            MaxActivitesInfFlows(a1,a2)=1;
        end
    end
end
end
end

```

Appendix 1.16 RG.m

```

function [Color]=RG(b)
%% En färgskala med 101 olika RGB färger, från grönt->gult->rött + en grå färg
% Funktionen hämtar en färg för varje % (mellan 0.00 och 1.00) +
% möjligheten till grå färg på 1.0001.
if b<=1
    b=round(b*100)/100;
end
rgb=[1.0001 0.600 0.60 0.6 ; %Grå figur
1 1.0000 0 0 ;
0.99 1 0.01 0 ;
0.98 1.0000 0.0200 0 ;
0.97 1.0000 0.0400 0 ;
0.96 1.0000 0.0600 0 ;
0.95 1.0000 0.0800 0 ;
0.94 1.0000 0.1000 0 ;
0.93 1.0000 0.1200 0 ;
0.92 1.0000 0.1400 0 ;
0.91 1.0000 0.1600 0 ;
0.9 1.0000 0.1800 0 ;
0.89 1.0000 0.2000 0 ;
0.88 1.0000 0.2200 0 ;
0.87 1.0000 0.2400 0 ;
0.86 1.0000 0.2600 0 ;
0.85 1.0000 0.2800 0 ;
0.84 1.0000 0.3000 0 ;
0.83 1.0000 0.3200 0 ;
0.82 1.0000 0.3400 0 ;
0.81 1.0000 0.3600 0 ;
0.8 1.0000 0.3800 0 ;
0.79 1.0000 0.4000 0 ;
0.78 1.0000 0.4200 0 ;
0.77 1.0000 0.4400 0 ;
0.76 1.0000 0.4600 0 ;
0.75 1.0000 0.4800 0 ;
0.74 1.0000 0.5000 0 ;
0.73 1.0000 0.5200 0 ;
0.72 1.0000 0.5400 0 ;
0.71 1.0000 0.5600 0 ;
0.7 1.0000 0.5800 0 ;
0.69 1.0000 0.6000 0 ;
0.68 1.0000 0.6200 0 ;
0.67 1.0000 0.6400 0 ;
0.66 1.0000 0.6600 0 ;
0.65 1.0000 0.6800 0 ;
0.64 1.0000 0.7000 0 ;
0.63 1.0000 0.7200 0 ;
0.62 1.0000 0.7400 0 ;
0.61 1.0000 0.7600 0 ;
0.6 1.0000 0.7800 0 ;
0.59 1.0000 0.8000 0 ;
0.58 1.0000 0.8200 0 ;
0.57 1.0000 0.8400 0 ;
0.56 1.0000 0.8600 0 ;
0.55 1.0000 0.8800 0 ;
0.54 1.0000 0.9000 0 ;
0.53 1.0000 0.9200 0 ;
0.52 1.0000 0.9400 0 ;
0.51 1.0000 0.9600 0 ;
0.5 1.0000 0.9800 0 ;
0.49 1.0000 1.0000 0 ;
0.48 0.9800 1.0000 0 ;
0.47 0.9600 1.0000 0 ;
0.46 0.9400 1.0000 0 ;
0.45 0.9200 1.0000 0 ;
0.44 0.9000 1.0000 0 ;
0.43 0.8800 1.0000 0 ;
0.42 0.8600 1.0000 0 ;
0.41 0.8400 1.0000 0 ;
0.4 0.8200 1.0000 0 ;
0.39 0.8000 1.0000 0 ;
0.38 0.7800 1.0000 0 ;
0.37 0.7600 1.0000 0 ;
0.36 0.7400 1.0000 0 ;
0.35 0.7200 1.0000 0 ;
0.34 0.7000 1.0000 0 ;
0.33 0.6800 1.0000 0 ;
0.32 0.6600 1.0000 0 ;
0.31 0.6400 1.0000 0 ;
0.3 0.6200 1.0000 0 ;
0.29 0.6000 1.0000 0 ;
0.28 0.5800 1.0000 0 ;
0.27 0.5600 1.0000 0 ;
0.26 0.5400 1.0000 0 ;
0.25 0.5200 1.0000 0 ;
0.24 0.5000 1.0000 0 ;

```

```
0.23 0.4800 1.0000 0 ;
0.22 0.4600 1.0000 0 ;
0.21 0.4400 1.0000 0 ;
0.2 0.4200 1.0000 0 ;
0.19 0.4000 1.0000 0 ;
0.18 0.3800 1.0000 0 ;
0.17 0.3600 1.0000 0 ;
0.16 0.3400 1.0000 0 ;
0.15 0.3200 1.0000 0 ;
0.14 0.3000 1.0000 0 ;
0.13 0.2800 1.0000 0 ;
0.12 0.2600 1.0000 0 ;
0.11 0.2400 1.0000 0 ;
0.1 0.2200 1.0000 0 ;
0.09 0.2000 1.0000 0 ;
0.08 0.1800 1.0000 0 ;
0.07 0.1600 1.0000 0 ;
0.06 0.1400 1.0000 0 ;
0.05 0.1200 1.0000 0 ;
0.04 0.1000 1.0000 0 ;
0.03 0.0800 1.0000 0 ;
0.02 0.0600 1.0000 0 ;
0.01 0.0400 1.0000 0 ;
0 0.0200 1.0000 0 ];
[row,~]=find(rgb(:,1)==b);
color = rgb(row,2:4);
end
```

Appendix 1.17 SystemIteration

```

function
[ActivitiesCapacity,FlowCapacity,ActorActivities,TotalNrOfActivities,SysItActivities,SysItFlow
] = SystemIteration(t1,FlowType,Actors,InoperableActivities,InoperableFlows,MinMax)
% SystemIteration bygger på multiplikationsmetoden och sprider påverkan och
% beroenden i iterationer så länge någon kapacitetsförändring är större än
% 0,1%. SystemIteration Behöver veta vilka nedsättningar som finns
% ursprungligen. Den hämtar sedan mer data från CreateMatrix.

if nargin < 6 % För att låta valet av MinMax vara frivilligt (ingen input ger mean tillbaka)
    MinMax=[];
end
[InfOnFlow,DepOnFlow,ActorActivities,TotalNrOfActivities]=CreateMatrix(t1,FlowType,Actors,MinMax);
%1. Definiera startvärden utanför iterationer:
LenFlow= length(FlowType);
MaxFlowCapacity=ones(LenFlow,1);
MaxActivitiesCapacity=ones(1,TotalNrOfActivities);
InitFlowCapacity=ones(LenFlow,1)-InoperableFlows;
InitActivitiesCapacity=ones(1,TotalNrOfActivities)-InoperableActivities;
FlowCapacity= InitFlowCapacity;
ActivitiesCapacity= InitActivitiesCapacity;
PrevStepFlowCapacity= MaxFlowCapacity;
PrevStepActivitiesCapacity= MaxActivitiesCapacity;
FlowInfOnActor=zeros(LenFlow,TotalNrOfActivities);
ActivitiesInfOnFlow=zeros(LenFlow,TotalNrOfActivities);
RedActivities=zeros(1,TotalNrOfActivities);
RedFlow=zeros(LenFlow,1);
abort=0;
SysItFlow=MaxFlowCapacity; %spelar ingen roll vad de får för värde men måste ha rätt storlek
SysItActivities=MaxActivitiesCapacity; %spelar ingen roll vad de får för värde men måste ha
rätt storlek
Sysiteration=1000; %För att inte koden ska loopas oändligt länge sätts ett maxvärde.
for Iteration=1:Sysiteration
    if abort==0 % Om påverkan på samtliga flöde och aktiviteter är mindre 0.1% avbryts
funktionen
        %3. Flödets möjliga påverkan på aktören. Beräknas med flödets kapacitet, FlowCapacity,
och aktörens beroende, DepOnFlow.
        for a1=1:LenFlow
            for a2=1:TotalNrOfActivities
                FlowInfOnActor(a1,a2)= (PrevStepFlowCapacity(a1) - FlowCapacity(a1))*
DepOnFlow(a1,a2);
            end
        end

        %4. Beräkna aktörens kapacitet, ActorCapacity. Genom att dra bort den relativa
sänkning som FlowInfOnActor

        FlowInfOnActor=1-FlowInfOnActor;

        for a1=1:LenFlow
            for a2=1:TotalNrOfActivities
                RedActivities(a2)=prod(FlowInfOnActor(:,a2));
            end
        end
        RedActivities=1-RedActivities;

        ActivitiesCapacity = ActivitiesCapacity- RedActivities.*PrevStepActivitiesCapacity;

        %6. Aktörens möjliga påverkan på flödet.
        for a1=1:LenFlow
            for a2=1:TotalNrOfActivities
                ActivitiesInfOnFlow(a1,a2)=(PrevStepActivitiesCapacity(a2)-
ActivitiesCapacity(a2))* InfOnFlow(a1,a2);
            end
        end

        %6.1 Sparar iterationes kapacitetsförändringar
        SysItFlow=PrevStepFlowCapacity-FlowCapacity;
        SysItActivities=PrevStepActivitiesCapacity-ActivitiesCapacity;

        PrevStepActivitiesCapacity =ActivitiesCapacity;
        %7. Definiera flödet innan påverkan.
        PrevStepFlowCapacity= FlowCapacity;

        ActivitiesInfOnFlow=1-ActivitiesInfOnFlow;
        for a1=1:LenFlow
            for a2=1:TotalNrOfActivities
                RedFlow(a1)=prod(ActivitiesInfOnFlow(a1,:));
            end
        end
        RedFlow=1-RedFlow;

        %8. Flödets kapacitet efter påverkan av aktören.
        FlowCapacity = FlowCapacity - RedFlow.*PrevStepFlowCapacity;

```



```
    if SysItFlow(<)<0.001
        if SysItActivities(<)<0.001
            abort=1;
        end
    end
end
end
if abort==0
    disp('systemet är inte stabilt innan 1000 iterationer och simuleringen avbryts därför')
end
end
```