



CHALMERS



GÖTEBORGS UNIVERSITET

Utveckling av klimatpolitisk beräkningsteknik med sekventiella beslutsproblem under osäkra förhållanden

Kandidatarbete vid Data- och informationsteknik

David Ek
Thea Johansson
Gabriel Lindkvist
Isak Mesihovic
Adam Moström
Edvard Nauc ler

Institutionen f r Data- och informationsteknik
CHALMERS TEKNISKA H GSKOLA
G TEBORGS UNIVERSITET
G teborg, Sverige 2025

Kandidatarbete 2025

Utveckling av klimatpolitisk beräkningsteknik med sekventiella beslutsproblem under osäkra förhållanden

David Ek

Thea Johansson

Gabriel Lindkvist

Isak Mesihovic

Adam Moström

Edvard Naulér



GÖTEBORGS
UNIVERSITET



CHALMERS

Institutionen för Data- och informationsteknik
Chalmers tekniska högskola
Göteborgs universitet
Göteborg, Sverige 2025

Utveckling av klimatpolitisk beräkningsteknik med sekventiella beslutsproblem under osäkra förhållanden

David Ek Thea Johansson Gabriel Lindkvist Isak Mesihovic Adam Moström Edvard Nauclér

© David Ek, Thea Johansson, Gabriel Lindkvist, Isak Mesihovic, Adam Moström, Edvard Nauclér 2025.

Handledare: Patrik Jansson, Institutionen för Data- och informationsteknik

Rådgivare: Nicola Botta, Potsdam Institute for Climate Impact Research

Examinatorer: Patrik Jansson och Arne Linde, Institutionen för Data- och informationsteknik

Rättande lärare: Wolfgang Ahrendt, Institutionen för Data- och informationsteknik

Kandidatarbete 2025

Institutionen för Data- och informationsteknik

Chalmers tekniska högskola och Göteborgs universitet

SE-412 96 Göteborg

Telefon +46 31 772 1000

Formgiven i L^AT_EX

Göteborg, Sverige 2025

Utveckling av klimatpolitisk beräkningsteknik med sekventiella beslutsproblem under osäkra förhållanden

David Ek, Thea Johansson, Gabriel Lindkvist, Isak Mesihovic, Adam Moström, Edward Nauc  r

Institutionen f  r Data- och informationsteknik
Chalmers tekniska h  gskola och G  teborgs universitet

Sammandrag

Klimatkrisen medför komplexa beslutsutmaningar som sträcker sig över lång tid och präglas av stor osäkerhet. I denna kandidatuppsats undersöks en metod för sekventiellt beslutsfattande inom klimatpolitik, där beslut tas iterativt över tid under osäkra förhållanden. Arbetet bygger på en befintlig modell av Botta et al. [1] som kvantifierar vilka klimatbeslut som spelar störst roll under osäkerhet. Implementeringen av modellen var ursprungligen skriven i det funktionella programmeringsspråket Idris, men för att öka tillgängligheten har modellen översatts till de mer etablerade språken Haskell och Python. Översättningen möjliggjorde även en validering av modellens korrekthet och prestanda.

Denna rapport redogör för den teoretiska grunden för sekventiellt beslutsfattande och klimatpolicy. Vidare beskrivs två klimatmodeller, både den skapad av Botta et al. [1] som översattes och en vidareutvecklad modell, samt metodiken för kodöversättning. Resultaten visar att de översatta modellerna producerar resultat som är identiska med originalimplementationens utfall och att prestandan är bättre än den ursprungliga implementationen. Vidare diskuteras potentiella förbättringar av modellens effektivitet och användbarhet, såsom möjligheten att utöka modellen. Slutligen behandlas de samhällsliga implikationerna av klimatpolitiska beslut och det diskuteras kring hur förbättrade beslutsmodeller kan bidra till mer informativa och ansvarsfulla klimatstrategier.

Nyckelord: Dynamisk programmering, Funktionell programmering, Klimatpolitik, Kodöversättning, Modellering, Policy, Sekventiell beslutsfattning

Abstract

The climate change crisis presents complex decision-making challenges, many spanning extensive time horizons and characterized by much uncertainty. This thesis presents a method for sequential decision-making in climate policy, where decisions are made iteratively over time with uncertain information. The project builds upon an existing model by Botta et al. [1] which quantitatively measures which climate decisions matter the most. The implementation of the original model was written in the functional programming language Idris. To improve accessibility and longevity, the model was translated from Idris code to modern-day programming languages Python and Haskell. This also enabled validation of the model's correctness and testing of performance.

The report outlines the theoretical background in sequential decision-making and climate policy. Furthermore, it describes two climate-policy models, both the one developed by Botta et al. [1] and a further developed model, along with the procedures of translating code. The results show that the translated models reproduce the outcome of the original implementation in Idris. Potential improvements in efficiency and accessibility are highlighted, along with possibilities to extend the model with new case studies. Finally, the societal implications of climate policy decisions are discussed and how improved decision models may support more informed and responsible climate strategies.

Keywords: Dynamic programming, Functional programming, Climate policy, Code translation, Modelling, Policy, Sequential decision-making

Författarnas tack

Vi som genomfört detta arbete vill rikta ett stort tack till vår handledare Patrik Jansson för all hans hjälp med projektet. Vi vill även tacka Nicola Botta från Potsdam Institute for Climate Impact Research, för hans hjälp med att förstå SDP:er och *Matter Most*-projektets original-implementation.

David Ek, Thea Johansson, Gabriel Lindkvist, Isak Mesihovic, Adam Moström, Edvard Naucér, Göteborg, maj 2025

Lista över förkortningar

Nedan är en lista på förkortningar som används i denna rapport, listade i alfabetisk ordning:

DHU	Makrotillstånd karakteriserat av: Fördröjd omställning, hög ekonomisk nivå, klimatläge där allvarliga konsekvenser fortfarande kan undvikas.
DHC	Makrotillstånd karakteriserat av: Fördröjd omställning, hög ekonomisk nivå, klimatläge där allvarliga konsekvenser är oundvikliga.
DLU	Makrotillstånd karakteriserat av: Fördröjd omställning, låg ekonomisk nivå, klimatläge där allvarliga konsekvenser fortfarande kan undvikas.
DLC	Makrotillstånd karakteriserat av: Fördröjd omställning, låg ekonomisk nivå, klimatläge där allvarliga klimatkonsekvenser inte längre kan undvikas.
DP	Dynamisk programmering, <i>Dynamic Programming</i> .
IAM	Integrerad bedömningsmodell, <i>Integrated Assessment Model</i> .
SDP	Sekventiellt beslutsproblem, <i>Sequential Decision Problem</i> .
SHU	Makrotillstånd karakteriserat av: Startad omställning, hög ekonomisk nivå, klimatläge där allvarliga konsekvenser fortfarande kan undvikas.
SHC	Makrotillstånd karakteriserat av: Startad omställning, hög ekonomisk nivå, klimatläge där allvarliga konsekvenser är oundvikliga.
SLU	Makrotillstånd karakteriserat av: Startad omställning, låg ekonomisk nivå, klimatläge där allvarliga konsekvenser fortfarande kan undvikas.
SLC	Makrotillstånd karakteriserat av: Startad omställning, låg ekonomisk nivå, klimatläge där allvarliga konsekvenser är oundvikliga.

Nomenklatur

Nedan följer nomenklaturen för index, mängder, parametrar och variabler som har använts genomgående i modellformuleringen och analysen i denna rapport.

Index

t	Index för tidssteg
x	Index för tillstånd

Mängder

\mathbb{X}	Tillståndsrum där underrum $\mathbb{X}_t \subseteq \mathbb{X}$ är giltiga tillstånd vid tidssteg t
\mathbb{Y}	Beslutsrum där underrum $\mathbb{Y}_{t,x} \subseteq \mathbb{Y}$ är giltiga beslut i tidssteg t och tillstånd x

Parametrar

β	Diskonteringsfaktor
Δt	Tidsdiskretiseringssteglängd
T	Problemets tidshorisonts utsträckning (antal tidssteg)
$p(\cdot)$	Sannolikhetsparametrar för övergångar mellan tillstånd i modellen
π	Policy
N	Besluttidshorisont

Variabler

t	Tidssteg
x	Tillstånd
y	Beslut
ps, Π	Policysekvens

Innehåll

Sammandrag	i
Abstract	iii
Abstract	vi
Akronymer	vii
Nomenklatur	x
1 Inledning	1
1.1 Syfte	2
1.2 Forskningsöversikt	2
2 Teori	5
2.1 Dynamisk programmering och sekventiella beslutsproblem	5
2.2 Modellbeskrivning: Matter Most	7
2.2.1 Antaganden inom Matter Most-modellen	11
2.3 Klimatmodellens samhälleliga krav	12
2.4 Teori bakom vidareutveckling	12
2.4.1 Modellera osäkerhet med normalfördelningar	12

2.4.2	Beslutsproblem med flera intressenter eller beslutsfattare	14
3	Metod	17
3.1	Genomförande av översättning	18
3.2	Vidareutveckling	18
3.2.1	Implementering av nya SDP:er	18
3.2.2	Vidareutveckling av det teoretiska ramverket	20
3.3	Tekniska avgränsningar	20
4	Resultat	23
4.1	Verifiering av korrekthet, reproducerbarhet och prestanda	23
4.2	Resultat från vidareutveckling	25
5	Diskussion	27
5.1	Tolkning av resultat	27
5.2	Möjliga tillämpningsområden	28
5.3	Miljömässiga och samhälleliga aspekter	29
5.3.1	Samhälleliga krav	29
5.4	Avvikelser och begränsningar	30
5.4.1	Implementeringsegenskaper i Idris	31
6	Slutsatser	33
	Litteraturförteckning	36
	Appendix	i

1

Inledning

Klimatkrisen är ett av den moderna tidens mest komplexa problem, inte minst när det kommer till att ta beslut som påverkar framtida utsikter både sett till ekonomi och klimat. För att beslutsfattare ska agera rationellt och nå bästa möjliga resultat krävs datadrivet och välinformerat beslutsfattande. För att hantera beslutsfrågor i komplexa adaptiva system är analyser och beräkningsbaserade bedömningsmodeller användbara. Traditionella klimatpolitiska analyser blandar ofta integrerade bedömningsmodeller (*Integrated Assessment Models*, IAM) och scenarioplanering för att utvärdera policyalternativ och fatta beslut [2]. Dessa modeller fångar dock inte alltid karaktären av sekventiellt beslutsfattande, det vill säga beslut som måste tas successivt över tid och som kan anpassas i takt med att ny information framkommer.

Det går att bemöta utmaningarna med att fatta mer datadrivna och rationella beslut på flera sätt. Denna rapport utgår från ett ramverk inom kontrollteori och sekventiellt beslutsfattande, för att beräkna nyttan av olika policyer samt jämföra betydelsen av beslut. Sekventiellt beslutsfattande definieras av [3] som en situation där en beslutsfattare successivt observerar en process innan ett beslut fattas, ofta med en kostnad associerad till varje observation. Ett sekventiellt beslutsproblem (*Sequential Decision Problem*, SDP) formaliserar en sådan situation i termer av diskreta tidssteg och:

- en mängd möjliga tillstånd där systemet i ett givet tidssteg kan befinna sig;
- en uppsättning beslut som kan väljas i olika tillstånd;
- en övergångsfunktion som beskriver hur systemets tillstånd förändras givet ett visst beslut;
- en belöningsfunktion som kvantifierar värdet av att ta beslut och gå från ett tillstånd till ett annat.

Målet med sekventiellt beslutsfattande är typiskt tvådelat: (i) att formulera ett välgrundat SDP och (ii) att hitta en optimal policysekvens för att maximera en belöning eller minimera en kostnad [3]. En policysekvens är i detta sammanhang en serie policyer, där varje policy anger vilket beslut som ska fattas i varje möjligt tillstånd vid varje tidssteg. Längden på en policysekvens bestäms av den aktuella beslutshorisonten, det antal tidssteg som

analyseras. Den optimala sekvensen kan ofta bestämmas med dynamisk programmering enligt Bellmans optimalitetsprincip. I denna modell används en tidshorisont på 8-12 steg och sekvensen beräknas exempelvis med metoden *backwards induction*, där policyn för det sista tidssteget bestäms först och därefter förlängs sekvensen ett steg i taget bakåt i tiden.

En central styrka i ramverket är dess förmåga att inte bara identifiera vilka beslut som bör fattas, utan även när i processen de bör genomföras. Eftersom varje beslut påverkar framtida tillstånd, blir tidpunkten för beslutet ofta avgörande. Genom att kvantifiera beslutens konsekvenser över tid möjliggör modellen en analys av när i beslutssekvensen ett visst val har störst påverkan på det långsiktiga utfallet.

Kandidatarbetet tar sin utgångspunkt i ett specifikt SDP kopplat till klimatpolitik, beskrivet i Botta et al. [1]. Denna modell är en del av *Matter Most*-projektet, en samling publikationer som presenterar ett ramverk för sekventiellt beslutsfattande, modellering och beräknings-teknik [1, 4–7]. I Botta et al. [1] används ramverket för att analysera klimatpolitiska beslut, med fokus på att kvantifiera hur viktiga beslut är i relation till osäkra framtida tillstånd.

Det SDP som beskrivs i Botta et al. [1] modellerar beslutsprocesser kring växthusgasutsläpp, vilket är särskilt relevant med tanke på klimatkrisens långsiktiga effekter. Att fatta beslut under osäkerhet i detta sammanhang kräver modeller som kan hantera både komplexitet, långsiktighet och systempåverkan. Samtidigt är ramverket utformat så generellt som möjligt för att möjliggöra framtida användning även inom andra områden, exempelvis energiplanering, pandemihantering eller finansiell riskhantering.

1.1 Syfte

Syftet med projektet är att tillgängliggöra samt undersöka möjligheter till vidareutveckling av den implementering för sekventiellt beslutsfattande under osäkerhet som presenteras i Botta et al. [1]. För att uppnå detta avses den ursprungliga implementeringen, skriven i programmeringsspråket Idris, översättas till Haskell och Python. Bakgrunden till detta är tredelad. För det första lärs Idris inte ut i samma utsträckning som Haskell och Python, vilket begränsar den nuvarande implementeringens tillgänglighet. För det andra möjliggör en sådan översättning en validering av resultaten i Botta et al. [1] och ett säkerställande av metodens reproducerbarhet, vilka är centrala aspekter i den vetenskapliga metoden. För det tredje har underhåll av Idris 1 upphört, vilket ytterligare motiverar behovet av att överföra implementeringen till aktivt underhållna språk.

1.2 Forskningsöversikt

Forskningen om klimatpolitiska beslutsstöd har i årtionden dominerats av IAMs. En av de mest inflytelserika är DICE-modellen (*Dynamic Integrated Climate-Economy Model*) som introducerades av Nordhaus [8]. Senare versioner som DICE-2016R inkluderar vissa osäkerheter kring klimatkänslighet och ekonomisk tillväxt [9]. Trots sina styrkor bygger dessa modeller på förinställda utsläppsbanor och policyregler. De antar i huvudsak att

dagens beslut är bindande långt in i framtiden och att ny information endast hanteras via känslighetsanalyser. Detta betyder att de inte tar hänsyn till hur beslut kan anpassas över tid i takt med att ny information blir tillgänglig [9].

För att bättre fånga dessa aspekter har fokus ökat på adaptiv klimatpolicy, där klimatpolitiska beslut betraktas som en serie åtgärder över tid som successivt kan anpassas allt eftersom ny data eller förändringar observeras [10]. Denna adaptiva ansats innebär att besluten fattas under tydliga antaganden om framtida osäkerheter samt att strategin kontinuerligt uppdateras när ytterligare kunskap eller tekniska innovationer blir tillgängliga. Samtidigt som intresset för adaptiva strategier har vuxit, har även datavetenskaplig forskning gjort stora framsteg och utvecklat generella ramverk för SDP:er. Dessa ramverk kombinerar etablerade metoder från beslutsanalys med moderna algoritmer som dynamisk programmering och förstärkningsinlärning (*reinforcement learning*) [3].

Den modell som presenteras i Botta et al. [1] är ett steg bort från traditionella statiska modeller eftersom den ser beslutsfattande som en sekventiell process under osäkerhet. Däremot är den inte adaptiv i bemärkelsen att beslutsstrategin förblir fast och uppdateras ej då ny information tillkommer, utan besluten styrs av det tillstånd som redan är definierat. Framtida forskning strävar efter att utveckla klimatpolitiska beslutsmodeller som inte bara hanterar osäkerhet, utan dessutom anpassar sig över tid. Detta är i linje med principerna för adaptiv klimatstyrning som beskrivs av H. Lee et al. [2] och Shepherd [10].

2

Teori

I detta kapitel presenteras den teoretiska grunden till SDP:er och beslutsfattande inom klimatpolitik. Kapitlet inleds med en genomgång av vad som karaktäriserar ett SDP och generella lösningsmetoder till dessa. Därefter analyseras det SDP som definieras i Botta et al. [1] för att illustrera ett klassiskt exempel. Vidare utforskas hur ett SDP med ett mer nyanserat tillståndsrum kan konstrueras och hur flera konkurrerande mål inverkar på beräkningarnas utfall. Genomgående knyts teorin an till klimatforskning för att understryka metodens relevans och praktiska tillämpning.

2.1 Dynamisk programmering och sekventiella beslutsproblem

Kärnan i den teori som kandidatarbetet bygger på är beräkning av optimala strategier. Generellt finns det många optimeringsproblem där konkurrerande strategier ska utvärderas för att hitta en lösning som maximerar belöning eller minimerar kostnad. En generell metod för att lösa sådana problem är dynamisk programmering (hädanefter DP), som bryter ner processen i diskreta tidssteg och finner en lösning genom (ofta rekursiv) beräkning av ackumulerade belöningar.

Optimeringsproblem kan utgå från två slag av processer: deterministiska¹- eller stokastiska processer. Stokastiska processer, som är vanliga i bland annat finansiella och ekologiska system på grund av risk och osäkerhet, modelleras ofta som Markov-beslutsprocesser (MDP) [11, 12]. För att lösa dessa används en specialiserad form av DP, kallad stokastisk dynamisk programmering. Stokastisk dynamisk programmering har det specifika målet att maximera väntevärdet av ackumulerade belöningar.

För att vidare rama in karakteristiken hos en klass beslutsproblem finns en mall för SDP:er. I kapitel 1 konstaterades det att ett SDP består av fyra huvudkomponenter. Dessa definieras nu något mer ingående, som:

- ett tillståndsrum \mathbb{X} där underrummet \mathbb{X}_t innehåller alla giltiga tillstånd x för tidssteg

¹Deterministiska processer är inte av central karaktär i denna rapport och det går att ersätta stokastiska övergångar med deterministiska sådana i samfliga beskrivningar nedan. Av dessa två skäl behandlas inte det deterministiska fallet varken separat eller ingående.

t ;

- ett beslutsrum \mathbb{Y} där underrummet $\mathbb{Y}_{t,x}$ innehåller alla giltiga beslut y som kan väljas vid tid t och tillstånd x ;
- en övergångsfunktion $f(x_t, y)$ som, givet att man i tidssteg t befinner sig i tillstånd x_t och tar beslut y , genererar sannolikheten ² för att i tidssteg $t + 1$ hamna i tillstånd x_{t+1} , för alla tillstånd i \mathbb{X}_{t+1} ;
- en belöningsfunktion $r_t(x_t, y, x_{t+1})$ som är värdet av att ta beslut y i tillstånd x_t när nästa tillstånd (känt) är x_{t+1} .

Beräkning av optimala lösningar av SDP:er bygger på en central princip inom DP: Bellmans optimalitetsprincip. Formulerat som ett teorem innebär det att det optimala beslutet $y_t^* \in \mathbb{Y}_t(x_t)$ är beroende endast av det nuvarande tillståndet $x_t \in \mathbb{X}_t$ och framtida tillstånd, men inte av tidigare beslut eller beslutssekvenser [3, 13]. Inom stokastisk DP är modelleringen av den sekventiella beslutsprocessen som en Markovprocess vedertagen, och med detta följer att den optimala värdefunktionen är en lösning till Bellmanekvationen [14, 15]. Den generella formuleringen av Bellmanekvationen faller utanför denna rapportens avgränsning, men en intresserad läsare hänvisas till [16] för en mer utförlig genomgång.

Begreppet policy är av stor vikt inom beslutsfattningsproblem och det kopplar in beslutsagenter i det stokastiska DP-problemet. En policy $\pi : \mathbb{X} \rightarrow \mathbb{Y}$ är en funktion (en regel) som avbildar tillstånd på beslut. En sådan regel π_t säger hur en agent kommer att agera, det vill säga välja $y_t \in \mathbb{Y}_t$, i fallet att hon når tillstånd $x_t \in \mathbb{X}_t$ vid tiden $t \in \{0, \dots, T\}$. Som ekvation formuleras det

$$y_t = \pi_t(x_t).$$

En agent kan applicera flera policyer, då kallad för policysekvens, över loppet av en diskret Markovprocess med beslutshorisont $N \in \mathbb{Z}_{>0}$, vilket samlas i mängden $\Pi = \{\pi_t, \dots, \pi_{t+N-1}\}$. Inspirerat av [17] går det att, under milda antaganden, forma en lösningsmetod av Bellmanekvationen. För tidssteg $t + N, t + N - 1, \dots, t$ kan rådande SDP brytas ner i följande delproblem:

$$V_{t+N}(x_{t+N}) := r_{t+N}(x_{t+N}), \quad (2.1)$$

$$V_t(x_t) := \sup_{\Pi} \mathbb{E} \left[\sum_{k=t}^{t+N-1} \beta^{k-t} r_k(x_k, \pi_k(x_k), x_{k+1}) + \beta^N r_{t+N}(x_{t+N}) \middle| x_t \right], \quad (2.2)$$

där tillståndsövergångarna följer övergångsfunktionen $x_{k+1} = f(x_k, y_k)$. Här är $r_t : \mathbb{X}_t \times \mathbb{Y}_{t,x} \times \mathbb{X}_{t+1} \rightarrow \mathbb{R}$ belöningsfunktionen för associerade övergångar; $r_N : \mathbb{X}_N \rightarrow \mathbb{R}$ är den sista belöningen, som endast beror på det sista tillståndet $x_N \in \mathbb{X}_N$; $\beta \in [0, 1]$ är en fix, icke-negativ diskonteringsfaktor; funktionen $V_t : \mathbb{X}_t \rightarrow \mathbb{R}$ kallas den t :te värdefunktionen. Uttrycket i ekvation (2.2) innebär att den förväntade ackumulerade belöningen vid tid t är det maximala väntevärdet av summan av diskonterade framtida belöningar, adderat med

²Övergångsfunktionen kan antingen vara deterministisk (sannolikheten är ett för ett av tillstånden och noll för alla andra) eller stokastisk (flera tillstånd har nollskild sannolikhet, summan av alla sannolikheter är ett)

den slutliga belöningen, givet det nuvarande tillståndet x_t , där nästa tillstånd x_{k+1} dras från sannolikhetsfördelningen given av $f(x_k, y_k)$.

I beräkningsmetoder översätts Ekvation 2.1 vanligen till rekursiv form:

$$V_t(x_t) = \max_{y_t \in \mathbb{Y}_{t,x_t}} \mathbb{E} [r_t(x_t, y_t, x_{t+1}) + \beta V_{t+1}(x_{t+1})],$$

där $x_{t+1} = f(x_t, y_t)$. Belöningar kan påverkas genom beslut $y_t \in \mathbb{Y}_{t,x_t}$, där $y_t = \pi_t(x_t)$. Sekventiella beslutsproblem går ofta ut på att finna en optimal policysekvens. En optimal policysekvens $\Pi^* = \{\pi_t^*, \pi_{t+1}^*, \dots, \pi_{t+N-1}^*\}$ är en sekvens av beslutsregler som, för varje tid t och tillstånd x_t , väljer beslutet $y_t^* = \pi_t^*(x_t)$ som maximerar $V_t(x_t)$. Genom att jämföra utfallet av den optimala policysekvensen med utfallet från en sekvens som avviker från denna, går det att definiera mått för hur viktiga olika beslut är för problemets slutgiltiga utfall.

Frågan kvarstår om varför ett SDP är meningsfullt för att modellera en beslutsprocess för miljömässig omställning. Modeller är alltid approximationer, men SDP:er är en bra metod här. Främsta skälet är att den hanterar osäkerhet och dynamik, som vid klimatförändringar eller ny teknik, genom Markovprocesser. Det låter beslutsfattare optimera långsiktiga miljömål, vilket är avgörande för hållbarhet. Det går också att dra kopplingar till miljöpolicy och företagsekonomi. Beslut om utsläppsminskning och omställning fattas återkommande över tid, där varje val påverkar framtida möjligheter och baseras på tidigare åtgärder samt ny information om klimat och ekonomi [1, 18]. I företagsekonomisk investeringsanalys finns en parallell till den så kallade Markov-egenskapen. Denna egenskap innebär att beslut enbart baseras på det aktuella tillståndet, utan hänsyn till tidigare händelser. Detta påminner om det som på engelska kallas *sunk costs fallacy* - att redan gjorda investeringar inte bör påverka rationella framtida val. I båda fallen är det framtida utfall som väger tyngst. En sådan framtidsorienterad beslutslogik är central inte bara för affärsstrategi, utan även för effektiv klimatpolitik.

Nackdelen med metoden är dess beräkningsmässiga tyngd. Antalet möjliga tillstånd och beslut växer exponentiellt med problemets dimensioner, ett fenomen som Bellman benämnde 'komplexitetens förbannelse' [14]. Metodens verklighetsförankring är också helt beroende av hur belöningsfunktionen, tillståndsrummet och beslutsrummet är uppställda. Forskare gör fortfarande framsteg i hur dessa formuleras, till exempel använder [19] neurala nätverk och förstärkningsinlärning (*reinforcement learning*) för att estimerar belöningsfunktioner.

2.2 Modellbeskrivning: Matter Most

Det specifika SDP som presenteras i Botta et al. [1] (kallas hädanefter även för Matter Most-SDP:t) utgör en förenklad modell av klimatpolitiskt beslutsfattande. I SDP:t ställs en beslutsfattare vid ett antal tillfällen (bestämt av beslutshorisont $N \in \mathbb{Z}_{>0}$) inför valet att antingen starta en grön omställning eller att vänta med åtgärder. I varje tillstånd finns tre binära tillståndsvariabler som tillsammans beskriver världsläget. Ett tillstånd vid tidssteg t skrivs som $x_t = (x_t^{(1)}, x_t^{(2)}, x_t^{(3)})$, där varje tillståndsvariabel är binär. Kombinationerna av dessa tre variabler ger upphov till åtta möjliga tillstånd som beskriver

världens läge vid varje tidssteg. Den första variabeln $x_t^{(1)}$ beskriver om en omställning har påbörjats (S) eller om den har fördröjts till nästa tidssteg (D). Variabeln $x_t^{(2)}$ beskriver det ekonomiska läget, antingen högt (H) eller lågt (L) välstånd. Den sista variabeln $x_t^{(3)}$ anger om världen har fastnat i en utvecklingsbana mot allvarliga klimatförändringar (C) eller inte (U). En sammanställning av tillståndsvariablerna och samtliga tillstånd i Matter Most-SDP:t presenteras i Tabell 2.1.

$x^{(1)}$ (Policy)	$x^{(2)}$ (Ekonomi)	$x^{(3)}$ (Klimat)	
S = Started	H = High	U = Uncommitted	
D = Delayed	L = Low	C = Committed	

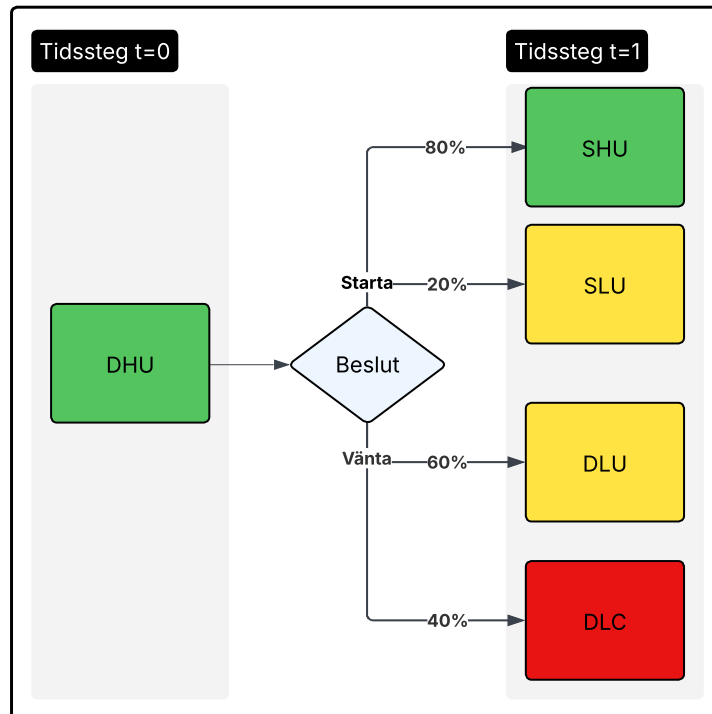
<i>Alla tillstånd (8 st):</i>			
SHU	SLU	SHC	SLC
DHU	DLU	DHC	DLC

Tabell 2.1: Översikt av tillståndsvariabler och alla möjliga tillstånd, som används i Matter Most-SDP:t.

I varje tillstånd där en klimatomställning inte startat har beslutsfattaren två möjliga alternativ: att försöka starta en grön omställning eller att avvakta. Inom Matter Most-SDP:t anses en startad omställning vara irreversibel, i de tillstånd där denna redan påbörjats har beslutsfattaren inga ytterligare beslut att välja mellan. För varje giltigt beslut som tas i ett givet tillstånd, genereras sedan en fördelning av övergångssannolikheter över de tillstånd systemet kan hamna i vid nästa tidssteg. Dessa sannolikheter beräknas genom funktionen next (motsvarande $f(x_t, y_t)$ i avsnitt 2.1) med hjälp av olika fasta parametrar beroende på systemets aktuella tillstånd och beslut. Till exempel ger startandet av en grön omställning initialt en ökad risk för att hamna i ett tillstånd med dålig ekonomi. Samtidigt medför tillstånd där en omställning inte är startad en ökad risk för att hamna i tillstånd där världen fastnat i en utveckling mot allvarliga klimatförändringar. Dessa exempel är delar av hur SDP:t försöker modellera samspelet mellan världens ekonomi och klimat.

De hittills beskrivna delarna av Matter Most-SDP:t - tillstånd, beslut och övergångar till nya tillstånd - bildar i sig själva en beslutsprocess. För att illustrera funktionaliteten i denna visas i Figur 2.1 en förenkling av ett steg i denna process. Figuren illustrerar hur en beslutsfattare, vid tidssteg $t = 0$, väljer mellan att starta en grön omställning eller att avvakta. Valet påverkar sannolikheten för att hamna i olika tillstånd i nästa tidssteg $t = 1$, där vissa tillstånd innebär klimatförsämring, låg ekonomisk aktivitet eller båda två. Den stokastiska strukturen i processen synliggörs då varje beslut leder till en sannolikhetsfördelad uppsättning utfall, snarare än ett bestämt resultat.

Första steget i ett beslutsträd för klimatpolitiskt SDP

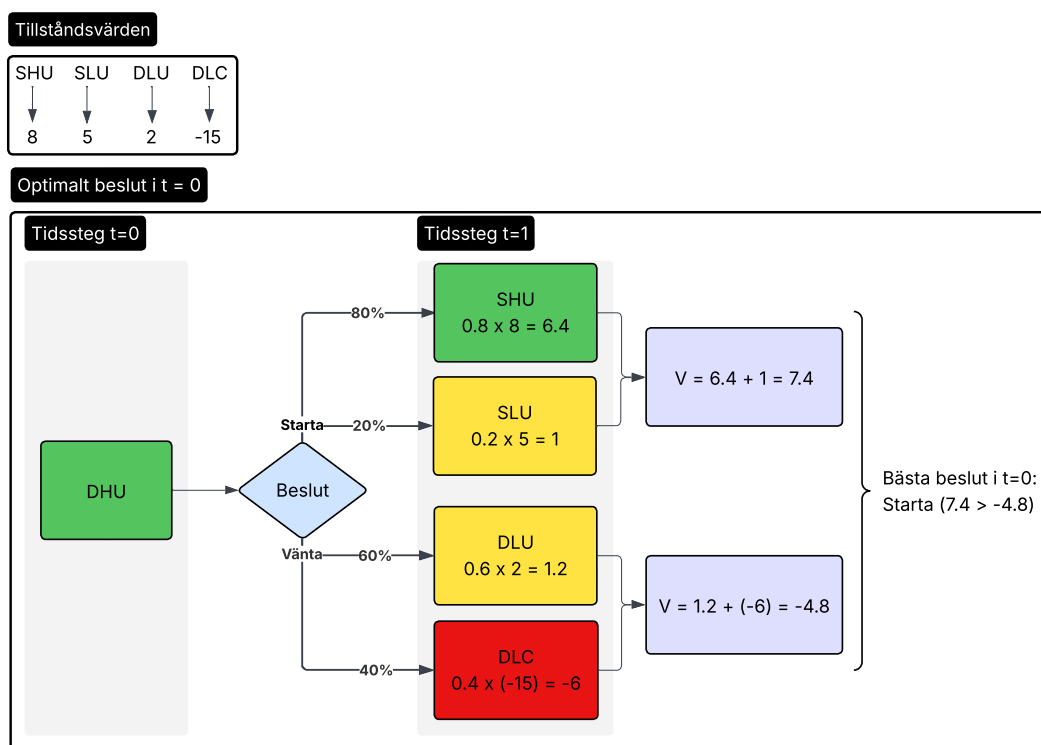


Figur 2.1: Illustration av beslutsträd i ett klimatpolitiskt sekventiellt beslutsproblem. DHU, SHU, SLU, DLU och DLC är exempel på olika möjliga tillstånd. Figuren visar hur olika beslut i ett tidssteg leder till olika sannolikheter för att i nästa tidssteg hamna i tillstånd med antingen bibehållen eller sämre miljö och ekonomi.

För att den hittills beskrivna beslutsprocessen ska bli ett beslutsproblem krävs ett mål, vilket formuleras med hjälp av belöningsfunktionen reward. Denna tar ett visst tidssteg t , ett tillstånd x_t , ett beslut y och ett efterföljande tillstånd x_{t+1} , och returnerar en belöning (detta motsvarar $r_t(x_t, y_t, x_{t+1})$ i avsnitt 2.1). I Matter Most-SDP:t tillämpas en belöningsfunktion som returnerar en 'poäng' varje gång $x_{t+1} \in (DHU, SHU)$, det vill säga varje gång systemet övergår till ett tillstånd med såväl hög ekonomisk nivå (H) som ett klimat inom vilket allvarliga förändringar fortfarande kan förhindras (U). Denna belöningsfunktion är grundstenen i processen att avgöra hur olika beslutsvärden utvärderas och därmed vilka sekvenser av policyer som är optimala.

De ovan nämnda delarna utgör tillsammans Matter Most-SDP:t som det beskrivs av Botta et al. [1]. Utöver dessa finns dock en uppsättning funktioner som används för att undersöka SDP:t. Den första av dessa är Va_1 , vilken beräknar det totala förväntade värdet av att från ett visst tillstånd x och vid ett givet tidssteg t följa en policysekvens ps (jämför med $V_t(x_t)$ i avsnitt 2.1). Funktionen summerar alltså belöningar över hela den för problemet bestämda beslutshorisonten. Figur 2.2 illustrerar hur värdet av två olika beslut i en

förenklad version av Matter Most-SDP:t skulle beräknas om det aktuella problemet hade en beslutshorisont $N = 1$.



Figur 2.2: Illustration av *backwards induction*. Beslutsfattaren utvärderar beslut bakåt i tiden utifrån förväntat framtida värde av $val(V)$.

För att hitta en sekvens av policyer som maximerar det förväntade värdet för ett SDP med en given beslutshorisont, används metoden baklängesinduktion. Detta innebär att man börjar med att beräkna den optimala policyn på den sista platsen i sekvensen. Därefter beräknas den bästa policyn på den näst sista platsen i sekvensen, givet att denna kommer att följas av den (nu beräknade) sista policyn. Denna process upprepas tills hela sekvensen beräknats. Varje enskild optimal policy beräknas med funktionen `bestExt`, som för ett tidssteg t och en policysekvens ps beräknar den optimala policyn för tidssteg $t - 1$, om den sedan följs av sekvensen ps . Detta görs genom att för varje giltigt tillstånd i tidssteg $t - 1$ - med hjälp av funktionen `Val` jämföra värdet av alla giltiga beslut i det aktuella tillståndet, vartefter det bästa sparas.

Den optimala policysekvensen beräknas i sin helhet med hjälp av funktionen `bi` som för ett tidssteg t och en beslutshorisont N rekursivt kallar på `bestExt` N gånger, först med tidssteg $t + N$ och sedan successivt bakåt till tidssteg t . På detta vis byggs den optimala policysekvensen upp som en sekvens av optimala 'förlängningar'. Tillsammans utgör `bi` och `bestExt` en tillämpning av Bellmans optimalitetsprincip.

När en optimal policysekvens beräknats använder Botta et al. [1] ytterligare två funktioner

för att analysera dessa. Den första, best , returnerar för ett givet tidssteg t , beslutshorisont N och tillstånd x det optimala beslutet att välja i detta tidssteg och tillstånd, samt dess värde. Detta sker genom att med funktionen bi beräkna den optimala policysekvensen utifrån gällande premisser och sedan extrahera det beslut som därigenom kopplats till det aktuella tidssteget och tillståndet. Den andra funktionen mMeas , används för att mäta betydelsen av ett beslut. Den kvantifierar hur stor roll ett beslut vid tidssteg t och tillstånd x spelar för det slutliga förväntade utfallet av beslutsproblemet, givet en beslutshorisont N . Detta görs genom att på samma sätt som för best beräkna en optimal policysekvens av längd N . Sedan jämförs värdet av att helt följa denna policysekvens (startande i tillstånd x) med värdet av samma sekvens med modifikationen att det sämsta beslutet väljs i det första tidssteget. Skillnaden i värde mellan originalsekvensen och den modifierade sekvensen normaliseras till att vara mellan 0 och 1, där 0 innebär att besluten inte leder till någon skillnad i utfall, och 1 innebär maximal skillnad.

Dessa funktioner utgör den huvudsakliga strukturen för modellen i Botta et al. [1]. Med hjälp av dessa kan modellen ta fram vilka policyer som är mest effektiva för att nå önskade tillstånd i framtiden och analysera när i beslutsprocessen ett visst beslut har störst betydelse.

En sista aspekt av originalmodellen från Botta et al. [1] som bör lyftas är att den är skriven i det beroendetypsbaserade språket Idris. Detta för att inte bara formulera beslutsprocessen, utan även formellt bevisa att vissa funktioner uppfyller önskade egenskaper. Sådana funktioner benämns här som *bevisfunktioner*, och används för att säkerställa att modellens centrala komponenter faktiskt leder till optimala beslut i enlighet med modellens definitioner.

2.2.1 Antaganden inom Matter Most-modellen

Matter Most-modellen bygger på ett antal antaganden och förenklingar som är viktiga för att förstå hur modellen fungerar, vad den tar hänsyn till och inte, samt vad den är lämplig att användas till.

Ett antagande är att modellen bygger på fasta sannolikheter, eller mer tekniskt uttryckt så klassas processen som en stationär MDP. Det innebär att den inte tar hänsyn till osäkerhet i sannolikhetsfördelningarna, utan dessa antas vara givna och konstanta. På detta sätt hanteras slumpmässighet i utfall, men inte osäkerhet i vår kunskap om systemen. Dessutom är sannolikheterna i modellen inte grundade i empiriska data eller expertbedömningar. Detta är en vanlig kritik mot IAM:s och andra klimatmodeller som används som beslutsstöd [20]. För att modellen ska kunna fungera som stöd i verkligt beslutsfattande behöver sannolikheterna antingen motiveras bättre eller vara mer anpassningsbara.

Modellen antar även att besluten är binära. Det vill säga att beslutsfattaren antingen startar en grön omställning eller väljer att vänta, samt att detta beslut inte kan ändras senare. I verkligheten är klimatpolitiska beslut ofta mer gradvisa och kan justeras över tid [18], men här används förenklingen för att förtydliga hur beslut påverkar framtida utfall.

Ett annat antagande är att beslutsfattandet sker på global nivå utan att ta hänsyn till skillnader mellan aktörer eller regioner. Det finns dessutom inga intressekonflikter eller problem

kring samarbete inbyggda i modellen. Dessa förenklingar gör modellen lämplig för att undersöka frågor om ansvar och tidssteg för åtgärder, men mindre lämplig för att spegla verkliga politiska processer där olika aktörer har olika mål och möjligheter att agera [18].

2.3 Klimatmodellens samhälleliga krav

Eftersom modellen i detta arbete tillämpas på klimatrelaterade beslut, är det relevant att lyfta några samhälleliga krav som ofta ställs på modeller inom klimatpolitik. De viktigaste kraven enligt Shukla, Skea et al. [21] är öppenhet och transparens. Det bör vara tydligt vilka antaganden som görs och hur modellen fungerar. Detta gör det lättare att förstå och granska resultaten från modellen. Parisavtalet lyfter även vikten av att beslut ska fattas med hänsyn till hållbar utveckling och rättvisa mellan generationer [22]. En modell ska kunna visa hur beslut påverkar tillstånd i framtiden, så att det går att avgöra hur olika beslut kan öka eller minska risken för allvarliga klimatförändringar. Ett annat krav är att modellen ska kunna hantera osäkerhet och förändrade förutsättningar. Eftersom framtiden är osäker behöver modeller kunna ta hänsyn till flera möjliga scenarier [2]. Dessa krav diskuteras vidare i underavsnitt 5.3.1.

2.4 Teori bakom vidareutveckling

Den ursprungliga modellen innehåller flera förenklingar som begränsar dess förmåga att tillämpas i verkligt beslutsfattande. Två centrala begränsningar är att sannolikheterna för tillståndsovergångar är fixerade samt att besluten antas fattas av en enskild aktör. I detta avsnitt diskuteras därför teorin bakom möjliga vidareutvecklingar av modellen för att bättre hantera dessa begränsningar.

2.4.1 Modellera osäkerhet med normalfördelningar

I den ursprungliga Matter Most-modellen utgörs tillståndsmängden av kombinationer med tre binära tillståndsvariabler: startad eller fördröjd omställning, hög eller låg ekonomisk aktivitet samt möjlighet eller avsaknad av möjlighet att undvika allvarliga klimatkonsekvenser. Sannolikheterna för övergångar från ett tillstånd till ett annat givet ett visst beslut är fixerade. Dessa antaganden är vanliga i beslutsmodeller eftersom de förenklar analysen, men de begränsar modellernas förmåga att hantera osäkerhet. Framförallt kan sådana modeller hantera slumpmässighet i utfall, men inte osäkerhet i sannolikhetsfördelningarna, så kallad epistemisk osäkerhet [10]. Detta är särskilt problematiskt i system där kunskapen är begränsad eller förändras över tid.

Till följd av att Matter Most-modellen bygger på en mycket liten tillståndsmängd har den svårt att fånga upp så kallade *fat-tail risks*. Det vill säga extrema utfall som har låg sannolikhet men mycket stor påverkan. I klimatrelaterade sammanhang är detta särskilt relevant eftersom osäkerheten kring de allvarligaste scenarierna är stor, och traditionella modeller tenderar att underskatta deras betydelse [8, 20]. Denna typ av risk är svårt att representera med vanliga sannolikhetsfördelningar som snabbt går mot noll i 'svansarna', såsom normalfördelningen. Teoretiskt sett kan dock *fat-tail risks* inkluderas i sekventiella

beslutsmodeller. Ett sätt är att använda sannolikhetsfördelningar med 'tjocka svansar' som till exempel Pareto- eller lognormalfördelningar, där sannolikheten för mycket allvarliga utfall inte kan försummas [23]. Att hantera sådana risker kräver modeller som kan uttrycka osäkerhet både i utfall och i själva sannolikhetsfördelningarna.

Även om modellen fortfarande inte fångar *fat-tail risks* och epistemisk osäkerhet, går det att minska vissa förenklingar i hur tillstånden förändras över tid. Detta kan göras genom att låta de ekonomiska och klimatmässiga tillstånden anta fler än två diskreta värden för att simulera en mer gradvis utveckling. Detta gör det möjligt att beskriva mer gradvisa förändringar i tillståndet utan att modellen använder kontinuerliga variabler. I den ursprungliga modellen av [1] finns en sannolikhet att misslyckas med en omställning. I en vidareutvecklad modell går besluten alltid igenom och risken modelleras istället med Gaussisk drift, det vill säga som en normalfördelad variation. Det utökade SDP:t består av 4 huvudkomponenter. Dessa definieras som:

- Makrotillståndet $x = (x_{clim}, x_{econ})$ har två komponenter, som indikerar graden av koldioxidförening på en heltalsskala $k = 1, \dots, K, K \in \mathbb{Z}_{>0}$ respektive det ekonomiska välståndet på en heltalsskala $l = 1, \dots, L, L \in \mathbb{Z}_{>0}$. Ramverket tillåter en utvidgning till ett godtyckligt antal variabler och beslut, men i denna studie begränsas det medvetet till två variabler för att likna Matter Most- modellen.
- \mathbb{Y} är de tillgängliga besluten vid alla tillstånd och tider.
- en övergångsfunktion $f = \mathbf{P}(k, l)$ som härleds nedan; k, l svarar mot olika makrotillstånd.
- en belöningsfunktion $r(x_t, y)$ som är värdet av att ta beslut y i tillstånd x vid tidsteg t .

Efter att en åtgärd y har valts modelleras osäkerheterna som två normalfördelningar, en för varje tillståndsvariabel. Den kontinuerliga väntade förändringen beräknas som nuvärdet plus en driftparameter Δ_y som beror på vilken åtgärd som tas. Detta ger förväntade nya värden:

$$\mu_{clim} = x_{clim} + \Delta_{clim}, \quad \mu_{econ} = x_{econ} + \Delta_{econ}.$$

där x_{clim}, x_{econ} är nuvarande värden och $\Delta_{clim}, \Delta_{econ}$ är driftsparametrar som beror på vald åtgärd y . Utfallen anses sedan vara stokastiskt kring dessa värden enligt en normalfördelning. För att återgå till diskreta värden beräknas sannolikheten för att den kontinuerliga variabeln hamnar inom respektive intervall. Detta görs med hjälp av den kumulativa fördelningsfunktionen:

$$P_{clim}(k) = P_{clim}(k-0.5 \leq X < k+0.5) = \Phi\left(\frac{(k+0.5) - \mu_{clim}}{\sigma}\right) - \Phi\left(\frac{(k-0.5) - \mu_{clim}}{\sigma}\right),$$

där Φ är den kumulativa fördelningsfunktionen för en normalfördelning, $k = 1, 2, \dots, K, K \in \mathbb{Z}_{>0}$ är mängden diskreta tillstånd. Termen ± 0.5 gör att intervallet $(k-0.5, k+0.5)$ samlar alla mikrovärden som ska kvantiseras till tillståndet k , exempelvis hamnar varje värde mellan 1.5 och 2.5 i tillståndet $k = 2$, ändpunkterna sätts till $-\infty$ och $+\infty$ för att fånga

upp alla utfall. På motsvarande sätt fås $P_{econ}(\ell)$, för $\ell = 1, 2 \dots L, L \in \mathbb{Z}_{>0}$. Utifrån sannolikheterna $P_{clim}(k)$ och $P_{econ}(\ell)$ definieras $K \times 1$ kolonnvektorer enligt

$$\mathbf{P}_{clim}(k) = (P_{clim}(1), \dots, P_{clim}(K))^T, \quad \mathbf{P}_{econ}(\ell) = (P_{econ}(1), \dots, P_{econ}(L))^T.$$

Övergångsmatrisen av ordnade par fås genom den kartesiska produkten:

$$\mathbf{P}(k, \ell) = \mathbf{P}_{clim}(k) \times \mathbf{P}_{econ}(\ell)^T = \begin{pmatrix} (P_{clim}(1)P_{econ}(1)) & \dots & (P_{clim}(1)P_{econ}(L)) \\ \vdots & \ddots & \vdots \\ (P_{clim}(K)P_{econ}(1)) & \dots & (P_{clim}(K)P_{econ}(L)) \end{pmatrix}.$$

Här ges sannolikheten att hamna i det specifika makrotillståndet $x_{k,\ell}$ av uttrycket $x_{k,\ell} = P_{clim}(k)P_{econ}(\ell)$. Genom att applicera övergångsoperatormatrisen element för element på sannolikhetsmatrisen \mathbf{P}_t och därefter summera de resulterande delmatriserna, erhålls den uppdaterade fördelningen \mathbf{P}_{t+1} . Värdefunktionen utvärderas sedan precis som i den ursprungliga modellen för att bestämma den optimala policysekvensen Π^* och för att avgöra vilka tidpunkter som besluten har störst marginaleffekt.

2.4.2 Beslutsproblem med flera intressenter eller beslutsfattare

Sekventiella beslutsproblem utgår traditionellt ifrån att beslut tas av en central beslutsfattare. Detta innebär ofta en förenkling av verkligheten, inte minst inom klimatområdet där flera aktörer påverkar varandra. Nyare studier visar att traditionella klimatpolitiska modeller med en enda beslutsfattare riskerar att missa viktig dynamik som uppstår i interaktionen mellan aktörer [24].

När flera självständiga aktörer fattar beslut som påverkar varandra analyseras situationen ofta inom spelteori, där varje aktör betraktas som en egen beslutsfattare som optimerar sin egen belöning över tid, till skillnad från modeller där en enda beslutsfattare maximerar den totala belöningen för hela systemet. Klimatproblemet kan ses som ett exempel på allmänningens tragedi: i icke-kooperativa klimatspel tenderar varje lands egenintresse att leda till högre globala utsläpp än vad som är optimalt ur ett världsperspektiv, ofta resulterande i en Nash-jämvikt – ett stabilt tillstånd där ingen aktör kan förbättra sin situation genom att ensidigt ändra sitt beslut [25]. Detta kan orsaka friåkningsproblemet, där aktörer undviker omfattande klimatåtgärder i väntan på att andra ska agera först, vilket leder till ett suboptimalt utfall. Utan en överstatlig myndighet som kan tvinga fram samarbete måste klimatbeslut vara självuppfyllande, där det ligger i varje aktörs egenintresse att agera, givet att andra gör detsamma. *Mechanism design* fokuserar på att utforma institutioner eller incitament som gör samarbete till det rationellt bästa valet för varje aktör [26].

En förenkling av icke-kooperativa scenarion kan uppnås genom att betrakta de olika aktörerna inom beslutsproblemet som intressenter istället för renodlade beslutsfattare. Med intressenter menas att aktörerna kan ha olika mål sinsemellan, men att det över dem finns en ensam beslutsfattare som tar de faktiska besluten. Ett sådant scenario kan modelleras som ett SDP genom att ge varje intressent en egen belöningsfunktion beroende av systemets tillstånd, och sedan ge den ensamma beslutsfattaren en belöningsfunktion som i sig är någon form av avvägning mellan intressenternas belöningsfunktioner. För varje givet beslut

kan intressenternas respektive belöningar även jämföras för att identifiera Pareto-optimala utfall, alltså utfall där ingen intressent kan få det bättre utan att någon annan får det sämre.

3

Metod

Projektet utgick från en befintlig implementering av en modell för sekventiellt beslutsfattande, utvecklad i det funktionella programmeringsspråket Idris av Botta et al. [1]. Arbetet delades in i två huvudsakliga faser: en översättningsfas och en vidareutvecklingsfas. Under hela arbetet låg fokus på tre centrala krav: att modellen skulle vara korrekt, ha tillräcklig prestanda och vara enkel att förstå och vidareutveckla. Dessa krav påverkade hur arbetet lades upp och vilka val som gjordes under översättning och utveckling.

Korrektheten var det mest grundläggande kravet. Den översatta modellen skulle ge samma resultat som originalimplementeringen i Idris för motsvarande indata. För att verifiera att översättningen var korrekt användes både enhetstester och egenskapsbaserade tester (*property-based testing*). Syftet med egenskapsbaserade tester är att för en bred mängd indata säkerställa att funktionernas egenskaper uppfylls som förväntat, till exempel associativitet och idempotens. I detta arbete inkluderades även verifiering av att funktionen `bestExt` returnerar den mest optimala policyn utifrån värdet från funktionen `val`. Enhets- tester användes för att kontrollera att specifika implementeringar gav förväntade resultat i definierade testfall. Utöver detta användes en omfattande testsvit med hög täckning för att öka tilltron till att översättningen var korrekt. Funktionerna `best` och `meas` som utgör ett slags användargränssnitt, testades även genom ett fullständigt test (*exhaustive test*). Detta innebar att samtliga permutationer av indata mellan 0 och 8 utvärderades, och resultaten från Idris jämfördes med motsvarande implementeringar i Haskell och Python för att styrka översättningens korrekthet.

För att underlätta framtida användning och vidareutveckling lades stort fokus på att strukturera upp koden och göra den läsbar för andra. Koden delades upp i abstrakta (generiska) funktioner och specifika implementeringsfunktioner, vilket innebär att en generisk (teoretisk) del sammanställer funktionsdeklarationer för de funktioner som är gemensamma för samtliga implementeringar, medan varje implementering av teoridelen (gränssnitt) utgör sina egna funktionsdefinitioner.

Hela kodbasen lades upp som öppen källkod på en publik *GitHub-repository* med referens till Botta et al. [1] och dokumentation över alla större ändringar. För mer information om dessa filer, se Bilaga 6.

3.1 Genomförande av översättning

Inledningsvis gjordes ett försök att översätta koden från Botta et al. [1] direkt, rad för rad till Haskell och Python. Ambitionen var att följa kodens struktur linjärt och återskapa varje funktionalitet på motsvarande sätt i det nya språket.

Under arbetets gång visade det sig dock att denna strategi var svår att genomföra i praktiken. Idris är ett beroendetypat språk där logik, typs-system och bevisföring är tätt integrerade. Detta gjorde att stora delar av koden inte kunde överföras direkt utan att antingen tappa funktion eller bli svåra att begripa i det nya språket. Efter ett tag byttes strategi.

Istället analyserades varje funktion separat för att identifiera dess syfte och matematiska innebörd. Funktionen implementerades sedan idiomatiskt i Haskell respektive Python, med hänsyn till specifika konventioner och praxis för de olika språken. På detta sätt kunde både läsbarhet och struktur förbättras, samtidigt som funktionalitet och logik bevarades.

Eftersom Idris till stor del använder formella bevis för att säkerställa korrekthet, ersattes dessa i den nya implementeringen med ett omfattande testsystem. Detta tillvägagångssätt gav större flexibilitet, samtidigt som det möjliggjorde en strukturerad och verifierbar översättning som låg i linje med projektets krav på tillförlitlighet och funktionell motsvarighet.

Strukturen i den översatta koden utformades för att möjliggöra vidareutveckling och breddad tillämpning. Detta omfattar dels implementering av andra typer av sekventiella beslutsproblem än det som behandlas i Botta et al. [1], dels modellutvidgningar såsom fler tillståndsvariabler, utökade valmöjligheter eller flera intressenter. För att identifiera relevanta utvecklingsmöjligheter genomfördes en litteraturgenomgång. Nya funktioner integrerades därefter och testades på samma sätt som grundmodellen för att säkerställa korrekt funktion.

3.2 Vidareutveckling

Utöver översättningen av den ursprungliga modellen har olika typer av vidareutvecklingar genomförts. Nya typer av SDP:er har implementerats inom det befintliga ramverket för att öka förståelsen för dess struktur och undersöka dess generaliserbarhet. Ramverket har dessutom utvecklats vidare, dels för att förbättra prestanda, dels för att möjliggöra scenarier med flera samverkande beslutsfattare.

3.2.1 Implementering av nya SDP:er

Utöver översättningen av det Matter Most-SDP som beskrevs i Botta et al. [1] implementerades tre andra testfall inom samma teoretiska ramverk.

Först implementerades ett enkelt 'tallinje-SDP'. Detta SDP utgick från ett antal tillstånd som representeras av heltal. I alla tillstånd på denna tallinje är de möjliga besluten att antingen gå nedåt, stå stilla eller gå uppåt utefter tallinjen. Belöningsfunktionen gav en

belöning som i alla lägen var identisk med tillståndet: $r(x_t, y_t, x_{t+1}) = x_{t+1}$. Den enkla belöningsfunktionen valdes för att det snabbt skulle gå att verifiera korrektheten i resultatet från beslutsproblemet.

Därefter implementerades ett mer generellt 'labyrinth-SDP', där tvådimensionella rutnät (kartor) föreställande labyrinth användes för att bestämma mängden giltiga tillstånd och beslut för dessa. Giltiga tillstånd gavs av alla rutor som inte var markerade som väggar, och giltiga beslut var att gå i alla riktningar vars angränsande ruta inte var markerad som vägg. Belöningsfunktionen gav en belöning då systemet befann sig i det tillstånd som ansågs vara 'utgången' av labyrinth. Vissa andra tillstånd var markerade som 'fällor' och gav en stor negativ belöning. Övriga giltiga tillstånd gav en liten negativ belöning (för att säkerställa att korta vägar ut ur labyrinth gav en högre belöning än långa).

Implementeringen av ett sista alternativt SDP genomfördes som en explorativ vidareutveckling av Matter Most-SDP, för att undersöka dess generaliserbarhet. Den teoretiska bakgrunden till detta utökade 'klimat-SDP' beskrivs i underavsnitt 2.4.1, där principerna för Gaussisk drift och diskretisering av normalfördelad osäkerhet presenteras. I vidareutvecklingen av modellen utökades de ekonomiska och klimatrelaterade tillståndsvariablerna från 'U/C' och 'H/L' till $k = 1, \dots, 5$ och $\ell = 1, \dots, 5$, för att stimulera en mer gradvis utveckling. Beslut antas alltid genomföras, och risken modelleras med Gaussisk drift följt av en normalfördelad osäkerhet. Tre åtgärdstyper definierades: ekonomisk, klimatomfattad och neutral. Ekonomiska åtgärder förväntades öka den ekonomiska tillståndsvariabeln, klimatåtgärder förväntades öka klimat-tillståndsvariabeln, medan den neutrala åtgärden inte förväntades ha någon påverkan på variablerna. De två tillståndsvariablerna antogs vara negativt korrelerade, vilket innebär att en ökning i den ena tenderar att sammanfalla med en minskning i den andra. Detta modellerades med hjälp av driftparametrarna, ett exempel på värden som testades kan ses i Tabell 3.1 nedan.

Åtgärd	Δ_{econ}	Δ_{clim}
Ekonomi	+1.0	-0.5
Klimat	-0.5	+1.0
Neutral	0	0

Tabell 3.1: Exempel på driftparametrar för respektive åtgärd. Driftparametrarna visar vad som väntas hända med medelvärdet i övergången till nästa tillstånd.

Övergångsoperatören med sannolikhetsfördelningar implementerades och testades i både Python och Haskell. Generellt användes en normalfördelad osäkerhet och samma belöningsfunktion som i det ursprungliga Matter Most-SDP:t. För att fånga extremutfall prövades även andra fördelningar, såsom exponentiella fördelningar. Alternativa belöningsfunktioner med förstärkt riskaversion undersöktes också, exempelvis $r = \frac{x_{\text{clim}}}{K} \cdot \frac{x_{\text{econ}}}{L}$.

Implementeringen av 'tallinje-SDP:t' och 'labyrinth-SDP:t' genomfördes endast i Python på grund av begränsningar i tid. Det 'utökade klimat-SDP:t' implementerades både i Python och Haskell. På grund av beräkningarna av normalfördelningar krävde detta SDP större beräkningsresurser än Matter Most-SDP:t, men det kunde fortfarande användas till att

beräkna policyer.

3.2.2 Vidareutveckling av det teoretiska ramverket

Vid sidan av implementeringen av alternativa SDP:er inom den ursprungliga beräkningsmodell som översatts från [1, 4–7] genomfördes två utvecklingar av detta ramverk. Den första var en implementering av memoisering vid beräkning av optimala policyer, och den genomfördes för att förbättra modellens prestanda (se kapitel 4). Memoisering innebär att resultat från tidigare funktionsanrop sparas i en cache baserat på indata. Vid framtida anrop med samma parametrar kan funktionen återanvända det sparade resultatet istället för att göra en ny beräkning. Detta minskar antalet redundanta beräkningar, särskilt i funktioner med överlappande delproblem, som ofta uppstår i rekursiva algoritmer. Memoiseringen implementerades i både Python och Haskell.

Den andra utvecklingen av det teoretiska ramverket från [1, 4–7] gjordes för att tillåta SDP:er med flera intressenter på det sättet som beskrevs i underavsnitt 2.4.2. På grund av tidsbegränsning genomfördes denna utveckling enbart i Python. Grunden i utvecklingen var införandet av ett 'multi-SDP'-objekt som fylldes med ett 'beslutsfattare-SDP' och ett valfritt antal 'intressent-SDP:er'. Beslutsfattare-SDP:t fungerar utifrån det vanliga teoretiska ramverket från [1, 4–7], med skillnaden att dess belöningsfunktion kan göras beroende av belöningsfunktionerna från samtliga individuella intressent-SDP:er. Varje 'intressent-SDP' ärver i sin tur sina tillstånds-, besluts- och övergångsfunktioner från 'beslutsfattare-SDP:t', men har en egen belöningsfunktion. Till 'Multi-SDP'-objektet skrevs även funktioner som genererar slumpmässiga policyer och beräknar värdet av dessa för varje individuell intressent. I fall där två intressenter ska jämföras innebär detta att deras respektive värden för en mängd olika policys kan ritas in på en graf, varefter en approximativ Pareto-front kan ritas ut.

En implementering av denna teoretiska utveckling gjordes i form av en variant av Matter Most-SDP:t från Botta et al. [1]. I detta 'multi-intressent-SDP' introducerades två intressenter istället för att enbart ha en ensam beslutsfattare. Dessa två gavs varsin belöningsfunktion: Den första blev belönad varje gång systemet hamnade i ett 'U'-tillstånd, alltså ett tillstånd där jorden inte ännu var på en bana mot svåra framtida klimatförändringar. Den andra gavs en (lika stor) belöning varje gång systemet hamnade i ett 'H'-tillstånd, alltså tillstånd där jordens ekonomiska aktivitet klassades som hög. Ett stort antal slumpmässiga policysekvenser genererades. Intressenternas respektive värden för dessa beräknades och visualiserades tillsammans med en approximativ Pareto-front (se kapitel 4).

3.3 Tekniska avgränsningar

Översättningen av det modellramverk som beskrivs i Botta et al. [1] begränsades till språken Haskell (GHC 9.4.8) och Python (3.10). Endast de ramverk och bibliotek som krävdes för att implementera funktionaliteten i den ursprungliga implementeringen inkluderades. Om motsvarande verktyg fanns tillgängliga i de nya språken användes dessa i syfte att bibehålla kodens enkelhet, tydlighet och kompatibilitet. Inga ytterligare anpassningar gjordes för

andra språkversioner eller miljöer. Formella bevis och lemmafunktioner som ingick i Idris-implementeringen översattes inte, utan ersattes av omfattande testning. Formell bevisföring ansågs inte nödvändig för att uppfylla projektets mål, som i första hand var att översätta och validera modellen i andra programmeringsspråk.

Det specifika Matter Most-SDP:t översattes med samma struktur som i Botta et al. [1], för att underlätta jämförelse med originalkoden. Detta SDP har 8 tillstånd, där 1 eller 2 beslut kan tas beroende på vilket tillstånd systemet befinner sig i vid ett givet tidssteg. SDP:t bygger på fasta sannolikheter, är inte adaptivt samt hanterar inte ny information över tid. Dessa aspekter gällde även för de alternativa SDP:er som implementerades, även om det exakta antalet tillstånd och möjliga beslut varierade mellan de olika specificeringarna. Utveckling av modellen i syfte att inkludera flera aktörer begränsades till att modellera intressenter under en central beslutsfattare, då en utveckling till flera autonoma beslutsfattare hade inneburit betydande förändringar av grunden i det teoretiska ramverket från Botta et al. [1].

Dessa begränsningar innebär att modellen inte lämpar sig för att modellera mer komplexa sammanhang och tillämpningar med större osäkerhet eller dynamiska förutsättningar. Begränsningarna påverkar även vilka typer av osäkerhet modellen kan hantera, vilket diskuteras i underavsnitt 2.4.1.

4

Resultat

I detta kapitel presenteras de huvudsakliga resultaten från arbetet. Fokus ligger på att utvärdera om de översatta ramverken och deras respektive versioner av Matter Most-SDP:t är korrekta, har tillräcklig prestanda och fungerar i praktiken. Dessutom redovisas resultat från de vidareutvecklingar som genomförts för att bredda och testa modellens användbarhet. Resultaten återges i relation till projektets syfte och utgör en grund för den efterföljande diskussionen.

4.1 Verifiering av korrekthet, reproducerbarhet och prestanda

För att säkerställa att de två översättningar som genomförts utgjorde korrekta implementeringar av det teoretiska ramverket från [1, 4–7] testades alla funktioner i båda versionerna med hjälp av ett stort antal egenskapstester. Med egenskapstester avses tester där funktioner verifieras mot formellt definierade egenskaper, för ett stort antal unika indata. Samma typ av tester genomfördes även på samtliga alternativa SDP:er som implementerades (se avsnitt 4.2). Resultaten visas i Tabell 4.1.

Språk	SDP	Klarade test
Python	Matter Most	58/58
Python	Tallinje	57/58
Python	Labyrint	57/58
Python	Klimat-utökat	58/58
Haskell	Matter Most	17/17
Haskell	Klimat-utökat	17/17

Tabell 4.1: Resultat av egenskapstester utförda på varje individuellt SDP som skrivits inom kandidatprojektet.

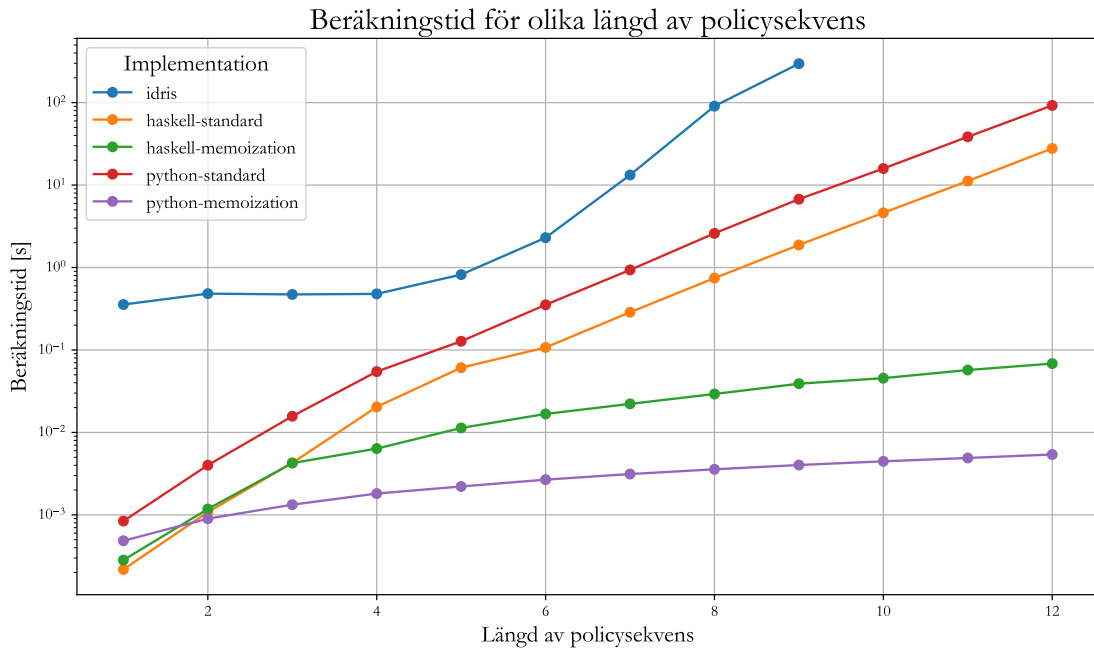
Som tabellen visar klarade både Matter Most-SDP:t och det utökade 'klimat-SDP:t' samtliga tester, både i sina Python- och Haskell-versioner. För 'tallinje-SDP:t' och 'labyrint-SDP:t', som båda implementerades i Python, misslyckades ett test. Detta test utvärderade beräkningen av egenskapen 'viktighet', som av Botta et al. [1] formulerats till att resultera i ett tal mellan 0 och 1. Ett sådant resultat förutsätter dock att SDP:t i fråga enbart returnerar

icke-negativa belöningar, något som inte är ett allmänt krav för sekventiella beslutsproblem. I 'tallinje-SDP:t' och 'labyrint-SDP:t' förekommer negativa belöningar, vilket gör att testet snarare påvisar en begränsning i jämförbarheten hos viktighetsmålet.

För att bekräfta att resultaten från Matter Most-SDP:t gick att reproducera testades Haskell- och Python-översättningarna av detta SDP, mot originalet i Idris av Botta et al. [1]. Resultaten visade att översättningarna gav identiska policyer och viktighetsmått (`mMeas`) och nästintill identiska nyttovärden som originalimplementeringen i Idris. Skillnaden i nytta var endast i storleksordning 10^{-15} , vilket kan ses som flyttalsavrundningar snarare än logiska avvikelser. Att resultaten stämmer trots att översättningarna inte gjordes rad för rad tyder på att den teoretiska modellens funktion har återskapats korrekt. Det kunde även bekräftas att resultaten från dessa tester var i linje med de slutsatser som presenteras i Botta et al. [1]. Tidiga klimatåtgärder är ofta optimala inom SDP:t, särskilt i lägen där det ekonomiska läget redan är stabilt. Dessutom tenderar vikten av beslut att vara störst i de första stegen av beslutsprocessen. Detta stärker både modellens reproducerbarhet och hur viktighetsmålet tolkas.

Prestandan hos de olika implementeringarna av modellramverket från [1, 4–7] jämfördes genom att mäta hur lång tid det tog att hitta en optimal policysekvens i Matter Most-SDP:t. Mätningen genomfördes för sekvenslängder mellan 1 och 12 steg¹. Resultaten som visas i Figur 4.1, visade att både Python- och Haskell-översättningen var snabbare än originalimplementeringen i Idris. Haskell-översättningen var dessutom något snabbare än den i Python. Figur 4.1 innehåller även resultat från motsvarande prestandatester från Haskell- och Python-översättningar med memoisering implementerad. För en mer ingående redovisning av dessa, se avsnitt 4.2.

¹Tidmätningen bör ses som en ungefärlig indikator, eftersom resultatet kan variera beroende på faktorer som processorkapacitet, belastning, schemaläggning med mera.



Figur 4.1: Graf som reflekterar prestandan för de olika versionerna där funktionen *best* körs med basala inputs ($t = 0$, $n = [1..12]$, $x = DHU$) i främsta syfte att mäta en övergripande prestandaförmåga. X-axeln visar längd på beräknad policysekvens, och y-axeln beräkningstid i sekunder med logaritmerad skala. Tidmätningen för Python och Idris använder Pythons *timeit*, och för Haskell används *System.CPUTime*.

4.2 Resultat från vidareutveckling

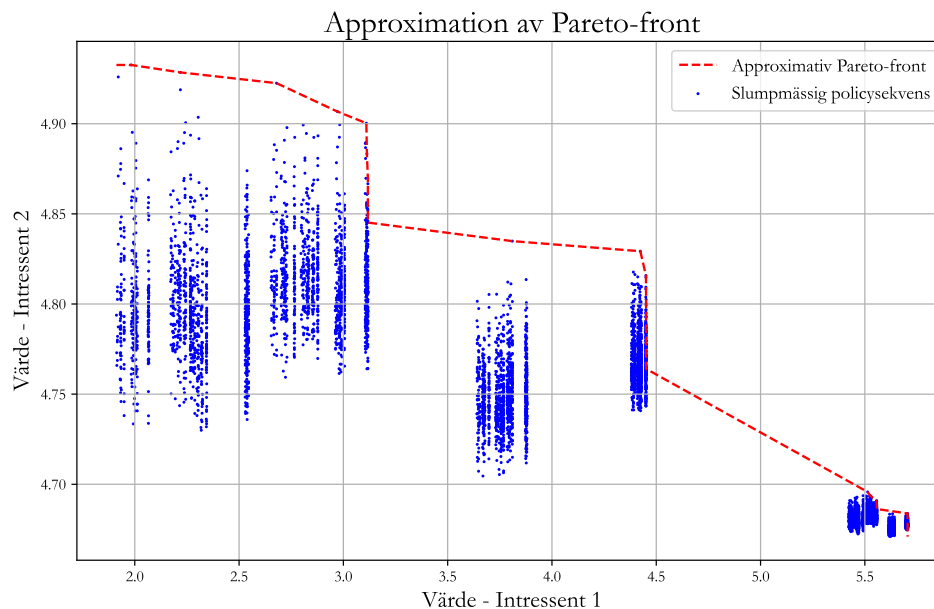
Utöver den rena översättningen av Matter Most-SDP av Botta et al. [1] genomfördes även tre typer av vidareutvecklingar. Dessa omfattade implementering av nya SDP:er inom det befintliga ramverket, samt två utvecklingar av själva ramverket: dels införande av memoisering för att förbättra prestanda, dels möjligheten att jämföra flera intressenter. Resultaten från dessa tre delar presenteras nedan i samma ordning.

Samtliga alternativa SDP:er som beskrivs i underavsnitt 3.2.1 kunde implementeras utan problem, och deras funktion verifierades med hjälp av de tester som redovisas i Tabell 4.1. Arbetet med att implementera dessa SDP:er i de översatta versionerna av modellramverket från Botta et al. [1] bidrog till att öka förståelsen för hur ramverket fungerar och vilka möjligheter och begränsningar det innebär vid modellering av SDP:er. Deras skilda syften visade tydligt att ramverket är generaliserbart. Som konstaterades i avsnitt 4.1, visade implementeringarna 'tallinje-SDP:t' och 'labyrint-SDP:t' också på en begränsning i hur det viktighetsmått som definieras av Botta et al. [1] kan användas.

Införandet av memoisering i både Python- och Haskell-översättningarna av modellen resulterade i en tydlig förbättring av prestandan vid beräkning av policyvärden och vid sökning efter optimala policysekvenser. Skillnaden jämfört med versionerna som ej använder memoisering framgår tydligt i Figur 4.1. Som nämns i underavsnitt 3.2.2 innebär memoise-

ring att tidigare beräkningar lagras och återanvänds, vilket undviker onödigt upprepade beräkningar. Det visade sig ha stor inverkan på prestandan i policyrelaterade beräkningar i SDP:er. I praktiken innebär detta att modellerna som använde memoisering kunde hantera beslutsproblem med betydligt längre beslutshorisonter än tidigare. Utan memoisering hade de längsta optimala policysekvenserna som kunde beräknas för Matter Most-SDP:t mellan 8 (Idris) och 12 (Python, Haskell) tidssteg innan beräkningstiden blev orimligt lång. Med memoisering kunde däremot en motsvarande policysekvens med 500 tidssteg beräknas på under en sekund.

Ramverket kunde också utvidgas för att hantera flera intressenter, vilket genomfördes framgångsrikt i Python. Implementeringen av ett 'multi-intressent-SDP' inom denna utvidgning gjorde att effekten av givna policysekvenser på olika belöningsfunktioner kunde undersökas. I Figur 4.2 visas resultatet av en sådan undersökning, där värdet av 10 000 slumpmässiga policysekvenser av längd 10 visas för två olika intressenter. Intressent 1 föredrar här tillstånd med god miljö, och intressent 2 föredrar tillstånd med god ekonomi (se underavsnitt 3.2.2 för en mer detaljerad beskrivning). Figuren visar tydligt att dessa mål delvis står i konflikt med varandra, de policyer som gynnade den ena intressenten hörde ofta till de sämsta för den andra. Trots detta går det att identifiera Pareto-optimala policysekvenser, där ingen av de två intressenterna kan få ett högre värde utan att den andra får ett lägre värde. Den röda streckade linjen i Figur 4.2, den så kallade Pareto-fronten, är dragen längs med dessa policysekvenser.



Figur 4.2: Graf som visar värdet av 10 000 slumpmässiga policysekvenser av längd 10 för två olika intressenter inom ett SDP. X-axeln visar värde för intressenten som bryr sig om god miljö, och y-axeln visar värde för intressenten som bryr sig om god ekonomi. Den streckade röda linjen visar den approximativa empiriska Pareto-fronten för detta SDP.

5

Diskussion

Samtliga testade implementeringar (Idris, Haskell, Python) gav identiska resultat inom maskinprecision, vilket visar att modellen är robust. Utöver att bekräfta korrektheten visar arbetet att verktyget kan göras mer användarvänligt och enklare att vidareutveckla. Genom ökad tillgång och öppenhet förbättras möjligheten till granskning och vidare forskning, vilket i sin tur stärker potentialen för modellen som beslutsunderlag. Nedan diskuteras resultatet av modellen, modellens begränsningar och möjliga förbättringar, därefter de samhällsenliga och etiska aspekterna av klimatpolitiska beslut.

5.1 Tolkning av resultat

Ett centralt resultat från simuleringarna var att beslutens påverkan, mätt med hjälp av viktighetsmålet (m_{Meas}) var som störst i de tidiga tidsstegen. Det överensstämmer med teoretiska förväntningar inom klimatpolitik, där tidiga åtgärder ofta har oproportionerligt stor betydelse för långsiktiga utfall. I modellen ledde beslut om tidig grön omställning till en påtagligt minskad sannolikhet att världen hamnar i ett tillstånd där allvarliga klimatkonsekvenser inte längre kan undvikas. Det bekräftar att modellen fångar en viktig aspekt, det är inte bara åtgärden som spelar roll utan även tajmingen.

Resultaten visade också att nyttan av att agera jämfört med att vänta var som störst i ekonomiskt stabila tillstånd som exempelvis SHU och SLU. Det antyder att samhällsekonomisk stabilitet är en viktig faktor i bedömningen av när det är optimalt att agera. Det här visar på värdet av funktionen m_{Meas} , som identifierar när ett visst beslut gör mest skillnad givet det tillstånd systemet befinner sig i. För att kontrollera att modellen fungerade likadant i de olika implementeringarna jämfördes resultaten från Haskell- och Python-översättningarna med originalet i Idris. Policyer och viktighetsmål stämde helt och skillnaden i nytta var 10^{-15} , vilket kan ses som flyttalsavrundningar snarare än logiska avvikelser. Detta tyder på att översättningarna är korrekta och att modellen fungerar som den ska oavsett språk.

När det gäller prestanda gjorde memoiseringen stor skillnad. Den gjorde det möjligt att beräkna policysekvenser på upp till 500 steg på under en sekund, vilket är betydligt fler än tidigare 8-12 steg. Det innebär att mer komplexa beslutssituationer kan hanteras, vilket ökar modellens praktiska användbarhet i framtida tillämpningar.

Införandet av spelteori och flera intressenter gav också tydliga resultat. När två aktörer har olika mål uppstår målkonflikter, vilket blev tydligt i exemplet som presenteras i avsnitt 4.2, där en intressent prioriterade miljö och den andra ekonomi. I Figur 4.2 syns det att policyer som gynnar den ena ofta är bland de sämsta för den andra. Genom att rita ut Pareto-fronten blev det tydligt var kompromisserna finns och hur svårt det kan vara att hitta policyer som är optimala för båda. Det påminner om hur det ofta ser ut i verkliga beslutsituationer, där olika aktörer har skilda mål och kompromisser är svåra men nödvändiga.

Att tallinje-SDP:t och labyrint-SDP:t kunde implementeras inom ramverket visar att det är tillräckligt flexibelt för olika beslutsproblem. Tallinje-SDP:et användes för att testa en enkel struktur med linjära tillstånd. Labyrint-SDP:t byggde på kartor i två dimensioner, där varje ruta motsvarade ett tillstånd och väggar bestämde vilka beslut som var möjliga. Tillsammans visade dessa implementeringar att modellen kan anpassas till olika typer av problem utan att ramverket behöver ändras. Samtidigt visade de också att viktighetsmålet m_{Meas} inte alltid går att tillämpa, särskilt i fall där belöningar kan vara negativa. Det framgår av testresultaten i Tabell 4.1, där dessa SDP:er inte klarade testet att viktighetsmålet skulle ligga mellan talen 0 och 1. Det beror på att definitionen av måttet i Botta et al. [1] förutsätter att alla belöningar är icke-negativa, vilket inte gäller i de här fallen. Detta visar att vissa delar av ramverket funkar bra i klimatmodellen, men inte nödvändigtvis i andra typer av beslutsproblem.

Vidareutvecklingen där tillståndsvariablerna delades in i 5 nivåer visade att modellen kan utvidgas för att hantera mer gradvisa förändringar i tillstånd utan att gå över till kontinuerliga variabler. Resultaten visade att modellen kunde hantera förändringen utan att ramverket behövde ändras. Samtidigt blev beräkningarna tyngre, vilket betyder att man får göra en avvägning mellan hur detaljerad modellen är och hur snabbt den går att köra.

5.2 Möjliga tillämpningsområden

I detta arbete har modellen tillämpats på beslutsfattande som rör klimatet, men den bygger på en generell metod för att analysera sekventiella beslut under osäkerhet [1]. Det innebär att den även kan användas i andra sammanhang där beslut fattas stegvis över tid, där tidigare beslut och osäkerhet påverkar tillståndet, och där det finns ett långsiktigt mål som eftersträvas. Modellen är särskilt användbar i situationer där tidiga beslut får stor betydelse för framtiden, eftersom den fokuserar på hur mycket ett beslut påverkar möjligheten att nå ett visst tillstånd.

Ett exempel är pandemihantering, där beslut om åtgärder som exempelvis nedstängningar eller fördelning av resurser fattas successivt under osäkerhet kring smittspridning och mutationer. Modellen skulle kunna användas för att identifiera vilka tidiga beslut som är mest avgörande för att undvika allvarliga konsekvenser. En begränsning är dock att modellen som utvecklats i detta arbete är att den inte fångar upp *fat-tail risks*, alltså händelser med låg sannolikhet men mycket stor påverkan som exempelvis plötslig smittspridning [21]. Sådana risker går i teorin att modellera genom att använda sannolikhetsfördelningar med 'tjocka svansar', men modellen i detta arbete använder sannolikhetsfördelningar där extrema utfall snabbt blir osannolika och fångar därför inte alltid upp dem. Även om man sätter ett högt

negativt värde på ett allvarligt utfall, påverkar det bara beslutet om sannolikhetsfördelningen inte gör utfallet alltför osannolikt. Ett annat område där modellen kan tillämpas är energiplanering, där investeringar görs stegvis och påverkas av osäkerhet kring teknik, efterfrågan och politiska beslut. Precis som i klimatfrågan handlar det om sekventiella beslut under osäkerhet, där tidiga beslut kan få stor betydelse för framtiden. Energisystem liknar också det beslutsproblem som modellen är utformad för att hantera eftersom investeringar ofta är irreversibla och sträcker sig över lång tid. Med hjälp av memoisering kan modellen dessutom köras över flera tidssteg, vilket gör den mer användbar i sammanhang där långsiktiga effekter är viktiga. Samtidigt kan modellen inte hantera flera aktörer och regionala skillnader, vilket är viktigt inom energiplanering. Med en spelteoretisk vidareutveckling som hanterar flera aktörer och konflikter mellan mål skulle modellen kunna vara ett användbart verktyg inom energiplanering.

Modellen kan även tillämpas inom finansiell riskhantering, där beslut om investeringar tas stegvis under osäkerhet kring hur marknaden utvecklas. Precis som i klimatfrågan handlar det om sekventiella beslut där tidiga val kan ha stor inverkan på framtida utfall. Samtidigt förändras har finansiella system ofta snabbare dynamik och risknivåer kan förändras på kort tid. Eftersom modellen antar fasta sannolikheter och inte är helt adaptiv, riskerar den att underskatta den typen av snabba förändringar. Genom att införa stokastisk variation i tillståndsövergångarna, som i en av modellens vidareutvecklingar, kan den delvis anpassas till mer osäkra och gradvisa förändringar. För att modellen ska fungera i finansiella sammanhang krävs dessutom att den kan hantera osymmetriska risker som till exempel lognormalfördelade utfall och att besluten uppdateras löpande när ny information tillkommer.

5.3 Miljömässiga och samhällliga aspekter

Klimatförändring är inte bara ett tekniskt eller vetenskapligt problem, utan i högsta grad ett samhällligt och etiskt problem. Därför är det viktigt att reflektera över hur denna modell relaterar till miljö och samhälle.

5.3.1 Samhällliga krav

För att modellen ska kunna fungera som ett användbart och trovärdigt stöd till beslut inom klimatpolitiken behöver den uppfylla ett antal samhällliga krav. Ett av de viktigaste kraven är öppenhet och transparens kring data och antaganden [21]. I detta arbete är alla antaganden tydligt dokumenterade och både kod och dokumentation finns tillgängliga. I och med att detta arbete bygger vidare på ett tidigare projekt, och att det finns stor utvecklingspotential, har det varit viktigt att möjliggöra för andra att granska och bygga vidare på modellen.

En annan aspekt är rättvisa mellan generationer. Det handlar om att beslut inte bara ska optimera kortsiktiga utfall utan även värna om framtida generationer och deras möjligheter [22]. Modellen som detta arbete bygger vidare på möjliggör att beslutsfattare kan ta hänsyn till detta genom att kvantifiera hur mycket enskilda beslut påverkar utfall i framtiden [1]. Genom att använda tillståndsvariabler som beskriver om världen hamnar i ett bundet (*committed*) eller obundet (*uncommitted*) tillstånd kan man med hjälp av modellen analysera

långsiktiga konsekvenser av beslut. Modellen kan dessutom identifiera vilka beslut som är mest avgörande för att nå ett valt mål i framtiden. På så sätt möjliggör modellen att beslutsfattare fattar beslut som tar hänsyn till både nuvarande och kommande generationer. Hur mycket man tar hänsyn till framtida generationer varierar dock beroende på vilket mål man väljer att optimera.

Modellen bör även vara anpassningsbar och flexibel. Då framtiden är osäker och förutsättningar snabbt kan förändras är det viktigt att modeller inkluderar sannolikheter [2]. Genom att införa sannolikheter och flera möjliga framtida tillstånd är modellen mer anpassningsbar än många traditionella modeller, som exempelvis DICE [9]. Dock är modellen ännu inte helt anpassningsbar, då den inte är adaptiv och anpassar beslut och policyer efter ny information som tillkommer med tiden [10]. Detta är dock en generell utmaning inom modellering av verkligheten, eftersom full anpassning och flexibilitet är svår att uppnå när framtiden alltid innehåller osäkerhet.

5.4 Avvikelser och begränsningar

Detta arbete utgår från samma modellstruktur som Botta et al. [1], med ett begränsat antal tillstånd, diskreta beslut och en ändlig tidshorisont¹. Detta gjorde det möjligt att genomföra översättningen och jämföra resultat, men innebär också att modellen är begränsad i sin förmåga att representera gradvisa åtgärder, mycket långsiktiga konsekvenser och anpassning över tid. Eftersom SDP:er arbetar inom ett fast antal tidssteg, med grovt förenklade tillstånds- och beslutsrum samt fasta sannolikheter för övergångar mellan tillstånd, är det svårt att fånga effekter som byggs upp och förändras över tid. I klimatrelaterade beslut, där osäkerheten är stor och effekterna ofta visar sig långt fram i tiden, är detta viktiga begränsningar.

Modellen fungerar bra för att analysera hur olika beslut påverkar framtida utfall och när i tiden ett beslut får störst betydelse, men den är inte tänkt att användas som ett direkt beslutsstöd. Att sannolikheterna för övergångar mellan tillstånd inte baseras på empirisk data och att hela världen ses som en enhetlig aktör är exempel på antaganden som begränsar hur väl resultaten kan tillämpas på verkliga beslutssituationer. Ett exempel på en sådan begränsning är att Matter Most-SDP:t inte fångar *fat-tails* eftersom de givna sannolikhetsparametrarna gör extrema utfall mycket osannolika. Det innebär att vissa allvarliga scenarier får för lite genomslag i beslutsregeln, även om de ges ett högt negativt värde. Resultaten är också känsliga för hur belöningsfunktionen är formulerad. I Matter Most-SDP:t returneras belöningar i tillstånd som undviker både allvarliga konsekvenser för klimatet och låg ekonomisk nivå. Om man istället hade valt mål som att enbart minimera kostnader, öka tillväxt eller fördela resurser rättvist mellan generationer, hade man sannolikt fått andra policyval som resultat.

¹ Detta gäller såväl de direkta översättningarna av Matter Most-SDP:t som de olika alternativa SDP:er och modellutvecklingar som implementerats.

5.4.1 Implementeringsegenskaper i Idris

I och med att ett av målen med arbetet var att tillgängliggöra modellen är det relevant att diskutera vilka egenskaper som förändrats när Idris har ersatts av andra språk. Idris är ett funktionellt språk som använder beroende typer, vilket gör det möjligt att formellt kontrollera att matematiska definitioner och funktioner är korrekt implementerade [1]. Koden innehåller bevisobjekt och lemmafunktioner, vilket innebär att flera samband verifieras automatiskt när programmet kompileras. Detta stärker modellens korrekthet, men förutsätter också att användaren behärskar Idris och förstår hur den typbaserade verifieringen fungerar i praktiken.

Samtidigt är Idris ovanligt utanför vissa forskningssammanhang, har begränsat verktygsstöd samt är på väg att ersättas av Idris 2. Dessa faktorer gör det svårare att använda eller vidareutveckla modellen. Därför har modellen i detta projekt översatts till både Haskell och Python, och den formella typkontrollen har ersatts med omfattande testning (se kapitel 4). Även om detta innebär att den formella verifieringen i Idris inte längre används, har modellen blivit mer tillgänglig för andra användare och enklare att vidareutveckla. Det innebär en tydlig avvägning, där projektets mål att göra modellen mer användbar har fått väga tyngre än att bevara möjligheten till formell verifiering i själva språket.

6

Slutsatser

Denna kandidatrapport har framgångsrikt översatt en avancerad klimatpolitisk beslutsmodell från Idris till Haskell och Python, med bibehållen korrekthet och förbättrad prestanda och tillgänglighet. Översättningen bekräftade att modellen som Botta et al. [1] utvecklades är reproducerbar. Genom noggranna tester har det verifierats att översättningarna är semantiskt ekvivalenta med originalkoden. Detta stärker förtroendet för både koden och de vetenskapliga slutsatserna som drogs i det tidigare arbetet.

Prestandautvärderingarna visar att Haskell- och Python-implementeringarna överträffar Idris-versionerna i hastighet. Kombinationen av två språk ger, utöver den ökade tillgängligheten, även större flexibilitet: Haskell är effektivt för tyngre beräkningar medan Python enkelt interagerar med andra verktyg. Den öppna tillgången till kodbasen och dess modulära struktur gör det ännu enklare för andra att granska, använda och vidareutveckla modellen.

Ur ett klimatpolitiskt perspektiv visar simuleringarna vilka beslut som en beslutsfattare bör lägga mest fokus på och betonar vikten av tidiga åtgärder även under betydande osäkerhet. Samtidigt påminner arbetet om de begränsningar som följer med alla förenklade modeller och att kvantitativa resultat behöver kompletteras med egna bedömningar, särskilt när det finns risk för extrema händelser som modellen inte fångar. Modellen fångar exempelvis inte dynamiska förändringar över tid eller *fat-tail risks*, vilket innebär att beslut kan behövas tas med större hänsyn till osäkerhet än vad modellen indikerar.

I detta arbete har modellen vidareutvecklats i flera avseenden, bland annat genom införandet av memoisering för att förbättra beräkningseffektiviteten samt en utvidgning för att hantera flera intressenter med olika belöningsfunktioner. Dessa förändringar visar att ramverket är tillräckligt flexibelt för att anpassas till mer komplexa beslutsproblem. Fortsatta utvecklingsmöjligheter innefattar införandet av adaptivitet, hantering av kontinuerliga tillståndsrum samt modeller för parameterosäkerhet och extrema utfall. Sådana utvidgningar skulle kunna öka modellens relevans i situationer där framtida förhållanden förändras snabbt eller där sannolikhetsfördelningarna är osäkra och snedfördelade.

Sammanfattningsvis har arbetet resulterat i både en färdig akademisk rapport och ett konkret verktyg som kan användas som grund för framtida forskning om klimatstrategier. Översättningen och de föreslagna förbättringarna utgör en bra grund för fortsatta studier. Klimatutmaningen kräver mångsidiga angreppssätt, och detta arbete visar hur datave-

tenskapliga metoder kan kombineras med klimatvetenskap för att fatta mer välgrundade beslut.

Litteratur

- [1] N. Botta et al. "Responsibility Under Uncertainty: Which Climate Decisions Matter Most?" I: *Environmental Modelling & Assessment* 28 (2023), s. 337–365. doi: 10.1007/s10666-022-09867-w.
- [2] J. Romero H. Lee et al. *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Tillgänglig: <https://www.ipcc.ch/report/ar6/syr/>. 2023. doi: 10.59327/IPCC/AR6-9789291691647.
- [3] A. Diederich. "Sequential Decision Making". I: *International Encyclopedia of the Social & Behavioral Sciences*. Utg. av Neil J. Smelser och Paul B. Baltes. Oxford, Storbritannien: Pergamon, 2001, s. 13917–13922. isbn: 978-0-08-043076-8. doi: <https://doi.org/10.1016/B0-08-043076-7/00636-7>.
- [4] N. Botta et al. "Contributions to a computational theory of policy advice and availability". I: *Journal of Functional Programming* 27 (2017), s. 1–52. doi: 10.1017/S0956796817000156.
- [5] N. Botta et al. "Sequential decision problems, dependent types and generic solutions". I: *Logical Methods in Computer Science* 13.1 (2017). doi: 10.23638/LMCS-13(1:7)2017.
- [6] N. Botta et al. "The impact of uncertainty on optimal emission policies". I: *Earth System Dynamics* 9.2 (2018), s. 525–542. doi: 10.5194/esd-9-525-2018.
- [7] C. Ionescu et al. *Type Theory as a Framework for Modelling and Programming*. Cham, Schweiz: Springer, 2018.
- [8] W. D. Nordhaus. *A Question of Balance: Economic Modeling of Global Warming*. New Haven, CT, USA: Yale University Press, 2008.
- [9] W. D. Nordhaus. "Projections and Uncertainties about Climate Change in an Era of Minimal Climate Policies". I: *American Economic Journal: Economic Policy* 10.3 (2018), s. 333–360. doi: 10.1257/po1.20170046.
- [10] T. G. Shepherd. "Storyline approach to the construction of regional climate change information". I: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 475.2225 (2019). doi: 10.1098/rspa.2019.0013.
- [11] Lucile Marescot et al. "Complex decisions made simple: A primer on stochastic dynamic programming". I: *Methods in Ecology and Evolution* 4.9 (2013), s. 872–884. doi: 10.1111/2041-210X.12082.
- [12] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, USA: John Wiley & Sons, 1994.

- [13] R. S. Sutton och A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, USA: MIT Press, 2018.
- [14] Peijun Guo. "Dynamic focus programming: A new approach to sequential decision problems under uncertainty". I: *European Journal of Operational Research* 303.1 (2022), s. 328–336. issn: 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2022.02.044>.
- [15] David Blackwell. "Discounted dynamic programming". I: *The Annals of Mathematical Statistics* 36.1 (1965), s. 226–235.
- [16] B. O'Donoghue et al. "The Uncertainty Bellman Equation and Exploration". I: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Stockholm, Sverige: PMLR, 2018, s. 3836–3845. url: <https://proceedings.mlr.press/v80/odonoghue18a.html>.
- [17] Mauro Gaggero, Giorgio Gnecco och Marcello Sanguineti. "Dynamic Programming and Value-Function Approximation in Sequential Decision Problems: Error Analysis and Numerical Results". I: *Journal of Optimization Theory and Applications* 156.2 (2013), s. 380–416. issn: 1573-2878. doi: 10.1007/s10957-012-0118-2.
- [18] Kunreuther H. et al. *Integrated Risk and Uncertainty Assessment of Climate Change Response Policies*. In: *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Tillgänglig: https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_chapter2.pdf. 2014.
- [19] Dimitri Bertsekas och John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.
- [20] E. A. Stanton F. Ackerman et al. "Fat tails, exponents, extreme uncertainty: Simulating catastrophe in DICE". I: *Ecological Economics* 69.8 (2010), s. 1657–1665. doi: 10.1016/j.ecolecon.2010.03.013.
- [21] J. Shukla, J. Skea et al. "Climate Change 2022: Mitigation of Climate Change". I: *Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2022, s. 85–87. doi: 10.1017/9781009157926.
- [22] United Nations Framework Convention on Climate Change. *Paris Agreement*. Tillgänglig: https://unfccc.int/sites/default/files/english_paris_agreement.pdf. 2015.
- [23] Martin L. Weitzman. "On modeling and interpreting the economics of catastrophic climate change". I: *The Review of Economics and Statistics* 91.1 (2009), s. 1–19. doi: 10.1162/rest.91.1.1.
- [24] Steve Keen et al. "Misrepresentation and exclusion: A study of economics in integrated assessment models for climate policy". I: *Globalizations* 16.5 (2019), s. 659–676. doi: 10.1080/14747731.2018.1502492.
- [25] Prajit K. Dutta och Roy Radner. "Self-enforcing climate-change treaties". I: *Proceedings of the National Academy of Sciences of the United States of America* 101.14 (2004), s. 5174–5179. doi: 10.1073/pnas.0400489101.
- [26] Jean Tirole. *Economics for the Common Good*. Princeton University Press, 2017.

Appendix

Projektets källkod finns tillgängligt på github.com enligt nedanstående länk:

`https://github.com/adammostrom/ruu-thesis-project-port.git`