Construction and Utilization of Mechanism-based Causal Models

by

Tsai-Ching Lu

B.S. and B.B.A., National Chengchi University

M.S., New York University

M.S., University of Pittsburgh

Submitted to the Graduate Faculty of

Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

FACULTY OF ARTS AND SCIENCES

This dissertation was presented

by

Tsai-Ching Lu

It was defended on

December 1, 2003

and approved by

Gregory F. Cooper, Associate Professor, Intelligent Systems Program

Milos Hauskrecht, Assistant Professor, Intelligent Systems Program

Krzysztof W. Przytula, Senior Research Scientist, HRL Laboratories

Committee Chairperson: Marek J. Druzdzel, Associate Professor, Intelligent Systems Program

Marek J. Druzdzel

CONSTRUCTION AND UTILIZATION OF MECHANISM-BASED CAUSAL MODELS

Tsai-Ching Lu, Ph.D.

University of Pittsburgh, 2003

This dissertation studies how the mechanism-based view of causality can assist in construction and utilization of causal models for decision support. The mechanism-based view of causality is based on the theory of causal ordering, proposed by Simon [1953], which explicates causal asymmetries among variables in a self-contained set of simultaneous structural equations. I extend the theory of causal ordering to explicate causal relations in under-constrained sets of structural equations. Considering under-constrained models as intermediate representations of one's understanding of decision problems, I demonstrate that a model construction process can be viewed as the process of assembling mechanisms from under-constrained models into self-contained causal models. I formalize the reversibility property of a mechanism to support changes in structure in causal models containing reversible mechanisms. I introduce algorithms for deliberating atomic actions when one considers manipulating a variable or releasing a mechanism to achieve a decision objective. In addition, I introduce the concept of search for opportunities which amounts to both identifying the set of policy variables and computing their optimal setting for a decision objective. Search for opportunities presents decision makers with a list of ranked interventions based on the value of intervention computation. I implement an interactive system called *ImaGeNIe* that supports mechanism-based model construction and utilization. I conduct subject experiments and find that *ImaGeNIe* can effectively assist users in constructing causal models for causal reasoning.

To my mom, Yixin, and Shangen

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Marek J. Druzdzel, for his invaluable guidance and generous encouragement through out the course of this research. Not only was I inspired by his ambitious yet practical style of research but also his enthusiasm and faith toward life. He taught me to have *secret* research weapons: do not follow the crowd, pay attention to details, aim high and start now, play idea with concrete examples, be persistent, and more. It is too much to describe how lucky and grateful I am as one of his advisees.

I would like to thank my committee members for their precious time, suggestions, and comments. Professor Gregory F. Cooper had asked me thought-provoking questions and offered his genuine comments to this work. Professor Milos Hauskrecht provided me with insightful comments. Dr. K. Wojtek Przytula was a wonderful mentor who showed me the industrial aspects of scientific research. I would also like to thank Professor Michael Lewis for his helpful discussions and supports for my experiment, and Professor Tze-Yun Leong and Roger Flynn for their interesting discussions in the early stage of this work.

Thanks to all DSL members whom I have had the privileges and pleasures to work with, especially Denver Dash, Hans van Leijen, Jeroen Bogers for great discussions about causality, and Daniel Garcia Sanchez and Tomasz Sowinski for helping me in the design and programming of *ImaGeNIe*. Thanks to Haiqin Wang, Yan Lin, Agnieszka Onisko, Adam Zagorecki, Randy Jagt, and Changhe Yuan for their friendships and supports.

Thanks to China Aviation Development Foundation at Taiwan for three years of funding which allowed me to freely explore my doctoral research directions. Thanks to my teachers and friends for my precious experiences at the Hua Hsing High School founded by Madame Chiang.

Finally, I would like to thank my mom, grandma, and brother for their love and patience. Thanks to my parents-in-law for taking care of my son for the last two years of my study. Special thanks to my wife for her understanding, encouragement, and love.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Motivation

Causality plays an important role in our thinking. We constantly ask ourselves why things happen as they are. Our understanding of the "why" enables us to hypothesize how we can change the world toward our advantages. For example, when facing an economic recession, economists debate about the possible causes of the recession and seek policies to stimulate the economy; when observing a drop of students' retention rate, board of education wonders if the curriculum should be changed; when an automobile accident happens, one questions if a more careful maneuver could have prevented it. If we believe that there is a reason behind every event we see, we can safely say that causal reasoning guides us through our journey. As the study of Artificial Intelligence aims at building systems that can reason with human intelligence and assists our daily activities, it is evident that causal reasoning will remain one of the central research topics of artificial intelligence.

Causal models based on structural equations emerged from genetics, econometrics, and social sciences [Wright, 1934; Haavelmo, 1943; Simon, 1953; Goldberger, 1972] and have become a dominant representation for supporting causal reasoning in Artificial Intelligence [Glymour and Cooper, 1999; Pearl, 2000; Spirtes *et al.*, 2000]. In the last decade, researchers have developed methods for deriving causal relationships from data with background knowledge [Pearl and Verma, 1991; Cooper and Herskovits, 1992; Spirtes *et al.*, 1993], providing causal interpretations to decision-theoretic models such as Bayesian networks and influence diagrams [Druzdzel, 1992; Pearl, 1993; Spirtes *et al.*, 1993], predicting effects of actions and policies [Pearl, 1993; Spirtes *et al.*, 1993; Balke and Pearl, 1995; Galles and Pearl, 1997], learning causal relationships from experimental or non-experimental data [Pearl, 1995; Spirtes *et al.*, 1995; Cooper, 2000], and generating causal explanations for observed events [Halpern and Pearl, 2001; Pearl, 2000]. Researchers have reported a wide range of applications of causal models in various scientific and industrial areas [Glymour and Cooper, 1999; Pearl, 2000].

A causal model of a system consists of a set of simultaneous structural equations, each of which represents a causal mechanism active in the system. Simon developed the theory of causal ordering that explicates causal asymmetries among variables in a causal model and represents causal relations in the model as a causal graph [Simon, 1953; Simon and Rescher, 1966; Simon, 1979; Simon and Iwasaki, 1988]. Causality derived by the theory of causal ordering is known as the mechanism-based view of causality. And causal graphs have played an essential role in identifying causal claims that are otherwise difficult to derive [Pearl, 1998]. A simple example of a causal model and its corresponding causal graph is a power train system of an automobile, which consists of two mechanisms: the mechanism acting between the engine (E) and the transmission (T), and the mechanism acting between the transmission (T) and the wheels (W). When the automobile engine is turned on, it acts on the transmission (T), which in turn acts on the wheels (W). The theory of causal ordering derives the following causal graph: $E \rightarrow T \rightarrow W$.

The quality of causal reasoning depends on the quality of the causal models. A model is requisite if it contains everything that is essential for solving the problem and no new insights emerge about the problem [Phillips, 1984]. Building a requisite model requires intuition and creativity, since the notion of requisiteness is subjective. Constructing requisite causal models is often a laborious task for domain experts. It is important to assist the process of causal model construction so that we can increase the quality of causal reasoning when facing complex systems. In this dissertation, I ask how the mechanism-based view of causality can assist the process of model construction.

The mechanism-based view of causality explicates causal relations within a postulated model. Although a causal relation between two variables is asymmetric in a model, researchers have reported that it can be reversed when a mechanism is embedded in different operational contexts [Simon, 1953; Wold and Jureen, 1953; Wold, 1954; Simon and Rescher, 1966; Druzdzel, 1992; Spirtes *et al.*, 1993; Pearl, 2000]. In the example of the power train system, when we drive a car down a hill, it is common practice to slow down the car by switching to a lower gear. According to the theory of causal ordering, we have the causal graph: $E \leftarrow T \leftarrow W$. When we compare this graph with the causal graph in the previous operational context, we see that causal relations among variables are reversed. Such modeling assumes that (1) a modeler has sufficient prior knowledge for predicting effects of actions and (2) effects of actions are represented by modifying the set of structural equations. Researchers in econometrics refer to such modeling as changes in structure

[Koopmans, 1950; Hood and Koopmans, 1953], namely modifying local structural equations that are brought about by interventions. When a manipulated variable is directly governed by an irreversible mechanism, changes in structure reduces to the arc-cutting semantic as reported in [Pearl, 1993; Spirtes *et al.*, 1993]. For example, the rain (R) can get us wet (W), $R \rightarrow W$; however, wearing the rain coat can only prevent us getting wet but it does not make the rain go away, i.e., the arc between R and W is cut, but not reversed. Whether a mechanism is reversible or not provides us with the knowledge necessary to determine which mechanism should be removed from the manipulated model. However, when manipulating on a system containing reversible mechanisms, no guidance has been provided in determining the effects of actions, namely which mechanisms should be invalidated and removed from the manipulated system. In this dissertation, I ask how the mechanism-based view of causality can support modelers in deliberating over changes in structure in systems containing reversible mechanisms.

When confronted with a complex system, decision makers may not know which variables one should best manipulate to achieve a decision objective, even given a requisite causal model that describes the system. Causal models can assist a decision maker in generating decision alternatives. I introduce the concept of search for opportunities which amounts to both identifying the set of policy variables and computing their optimal settings for a given decision objective. In this dissertation, I ask how to combine decision-theoretic methods with the mechanism-based view of causality to solve the problem of search for opportunities.

1.2 Statement of Thesis

The central thesis of this dissertation is as follows:

> *The mechanism-based view of causality provides an effective formalism for causal model construction and utilization.*

I study this thesis and demonstrate its correctness by extending the theory of causal ordering and by implementing a computer-aided interactive modeling environment *ImaGeNIe* for causal reasoning. In particular, I demonstrate that

Model Construction The model construction process can be viewed as the process of assembling

mechanisms from under-constrained models into self-contained causal models. The extended theory of causal ordering explicates causal relations in under-constrained models into causal graphs that can represent ones' intermediate understanding of decision problems.

Changes in Structure The reversibility property of a mechanism can support changes in structure in causal models containing reversible mechanisms. The formalization of reversibility leads to algorithms for deliberating atomic actions when one considers manipulating a variable or releasing a mechanism to achieve a decision objective.

Search for Opportunities The mechanism-based view of causality enables the computation of the value of intervention. The formalisms of the action operator **Act** and the augmented models allow the myopic search for opportunities to compute a sequence of intervening and non-intervening actions as decision alternatives.

1.3 Overview

This dissertation will be organized as follows:

Chapter 2 describes the theory of causal ordering and the graphical representation of causal models.

Chapter 3 presents the extended theory of causal ordering for under-constrained systems and shows how such theory can be used in assisting the process of model construction. An interactive and iterative modeling environment, *ImaGeNIe*, is presented to demonstrate the approach of mechanism-based causal model construction.

Chapter 4 formalizes the reversibility property of a mechanism and the **Act** operator for reasoning with systems containing reversible mechanisms. It presents two algorithms for deliberating atomic actions for changes in structure in reversible systems.

Chapter 5 introduces the concept of search for opportunities and the computation of the value of intervention. It demonstrates how to solve the problem of search for opportunities in causal models containing mixtures of mechanisms.

Chapter 6 describes the experimental results on using *ImaGeNIe* to construct causal models for causal reasoning.

Chapter 7 concludes the contributions of this dissertation and describes possible future research.

Chapter 2
Causal Models

A *system* is a piece of the real world that can be reasonably studied in isolation. A *model* describes the phenomena and mechanisms in a system. A model can be represented by a set of structural equations where each equation describes a conceptually distinct causal mechanism active in the system. Such models are known as Simultaneous Equation Models (SEMs) [Wright, 1934; Haavelmo, 1943; Simon, 1953], also called Structural Equation Models [Goldberger, 1972] in economics and social science. Simon [1953] developed the theory of causal ordering to explicate causal asymmetries among variables in a self-contained SEM. The theory of causal ordering provides a formal account of causality, known as mechanism-based view of causality.

This chapter reviews the theory of causal ordering and defines the concepts needed for the rest of this dissertation. Section 2.1 presents notations that will be used in this document. Section 2.2 and 2.3 describe the building blocks of causal models and discuss the assumptions practiced in structural equation modeling. Section 2.4 describes causal structure, the theory of causal ordering, and causal graph.

2.1 Notation

Variables will be denoted by capital letters, such as V, and values of variables by lower case letters, such as v. A set of variables will be written as upper case boldface type such as $\mathbf{V}$ and the values of a set of variables as lower case bolder face $\mathbf{v}$. $|\mathbf{V}|$ will denote the number of elements in the set $\mathbf{V}$. A graph $G = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ consists of a set of nodes $\mathbf{N}$, a set of directed arcs $\mathbf{A}$, and a set of undirected arcs $\mathbb{A}$. A graph $G = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ is (completely) directed if $|\mathbb{A}| = 0$, shorthanded as $G = \langle \mathbf{N}, \mathbf{A} \rangle$. A graph $G = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ is undirected if $|\mathbf{A}| = 0$, shorthanded as $G = \langle \mathbf{N}, \mathbb{A} \rangle$. The sets of parents, children, ancestors, and descendants of a node $N \in \mathbf{N}$ in a graph G are denoted by $\mathbf{Pa}(N)$, $\mathbf{Ch}(N)$, $\mathbf{Anc}(N)$, and $\mathbf{Des}(N)$ respectively.

2.2 Phenomena and Variables

Structural equation modeling starts with identifying entities involved in a system. An entity can be a single object (e.g., a patient), a population of similar objects (e.g., male patients in a hospital), or a group of relevant objects (e.g., patients, doctors, and insurance company in a health system). We define variables to represent phenomena of entities (e.g., the blood pressure of a patient) and define structural equations to describe mechanisms among the variables in a system. A set of structural equations that describes the system of interest is a SEM. Our prior domain knowledge decides which mechanisms are involved in a system. Therefore, the definitions of structural equations and variables in a SEM are a-priori [Simon, 1953; Wold, 1954].

Since a variable represents a phenomenon of an entity, we may associate different properties to a variable for different purposes of modeling. A variable has *value* as its default property. The value of a variable can be categorized into discrete or continuous, temporal or non-temporal, and spacial or non-spacial. When a variable is used in a statistical model, we may attach to the variable properties such as measured or unmeasured, and observable or non-observable. When a variable is used in experiments, we may attach to the variable properties such as manipulable or non-manipulable and dependent or independent. In general, a variable in a SEM should be treated analogically to a user-defined data type in a programming language, where a modeler has clear definition of what it represents in the real world and what properties it has.

2.3 Mechanisms and Structural Equations

Mechanisms are relations between phenomena, which are represented as structural equations over variables. When relations among phenomena are consistently observed, we postulate mechanisms to describe such relations. Rather than arbitrarily assert mechanisms among phenomena, we normally apply criteria such as *stability* to screen off *transitory* relations or favor one formalism over the other. For example, we say that causal relationships are more stable than probabilistic relationships because causal relationships ontologically describe physical relations among objects in the world, whereas probabilistic relationships epistemically reflect what we believe about our world [Pearl, 2000, pp. 25]. It is also stressed that a mechanism should be *autonomous* in the sense

that the external change on any one of the mechanisms in a system does not imply the change of others [Haavelmo, 1944, pp. 26]. For example, disconnecting the fuse that connects the headlights and the battery in an automobile does not imply that the generator will not generate power for the battery. A structural equation can be defined as follows.

Definition 2.1 (structural equation)
A structural equation $e(V_1, V_2, \ldots, V_n) = 0$ *represents an autonomous and stable mechanism among the set of variables* $\{V_1, V_2, \ldots, V_n\}$ *that represents the phenomena described by that mechanism.*

Simon [1979] pointed out that different *a-priori* assumptions for one structural equation may lead to different interpretations of causal relations among variables. For example, schooling helps in increasing verbal ability in one experimental context, but verbal ability helps in getting higher schooling in another. Simon used the term *causal mechanisms* to refer to mechanisms considered under different a-priori assumptions. In other words, a structural equation can be written in an explicit functional form to specify the effect and its causes in a-priori assumptions.

Definition 2.2 (structural equation in explicit form)
Let $e(V_1, V_2, \ldots, V_n) = 0$ *be a structural equation that represents a mechanism. Solution of this equation for* V_i*, i.e, the equation* $V_i = f_i(V_1, \ldots, V_{i-1}, V_{i+1}, \ldots, V_n)$ *is an explicit functional form of* e *that represents a causal mechanism where the set of variables* $\{V_1, \ldots, V_{i-1}, V_{i+1}, \ldots, V_n\}$ *are the causes of the effect variable* V_i*.*

In general, an equation is symmetric. Does it imply that a structural equation in its implicit form can be written into all its possible explicit functional forms and be interpreted causally? The answer is no. The assertions of causal relations require *a-priori* assumptions [Haavelmo, 1944; Simon, 1953; Wold, 1954]. Simon [1979; 1988] suggested three sources in asserting asymmetries: *manipulability*, *time precedence*, and *prepotency*.

Manipulability is based on the *empty world postulate* [Simon and Rescher, 1966] which states that

> "... most variables in the world are not directly connected with most other variables, and that such connections as exist involve a very small number of different kinds of mechanisms. Then, one would include a particular variable in a subsystem only if one

could select a mechanism from the list of admitted mechanism through which that variable could possibly act on that subsystem."

In other words, we can select one of admitted mechanisms in the system and manipulate on one of the variables in the selected mechanism such that the system will contain a mechanism that is linked to the manipulated variable alone. Manipulation in the context of randomized experimentation allows experimenters to test causal links between the independent variables and the dependent variable. However, when one takes the nature as the force of the manipulation and by this tests a mechanism from naturally occurring data, this postulate may be fallible when spurious links exist.

Time precedence is undoubtedly one of primary sources to conjecture asymmetries among variables. In the world we live, it is generally agreed that the effect cannot precede its causes in time. Consequently, we tend to apply regularity of succession to assert asymmetries among variables. Temporal precedence, however, does not always imply causal precedence since spurious correlation may be at play. We shall emphasize that time precedence is inessential for causality defined in the theory of causal ordering [Simon, 1953; Simon and Rescher, 1966; Simon, 1979; Simon and Iwasaki, 1988]. Nonetheless, explicit representations of temporal ordering among variables in a structural equation may assert causality explicitly.

Prepotency postulate states that phenomena with large or powerful force cause the small or weak phenomena as effects. For example, we normally say that Sun causes Earth to revolve, rather than the other way around. However, this postulate may be fallible when feedback mechanisms exist. For example, the amount of rains causally influence the amount of agricultural productions, but increasing the amount of agricultural productions by taking the land away from forests in the long run may influence weather in reverse. Prepotency postulate may also be fallible when control subsystems exist. For example, our nerve system controls our body movement although the mass of our nerve system is small, comparing to the mass of the rest of our body.

When a mechanism is embedded in different operational contexts, the causal asymmetries among variables may be reversed from one context to another. Such type of mechanisms is recognized as *reversible* mechanisms in [Simon, 1953; Wold and Jureen, 1953; Wold, 1954; Simon and Rescher, 1966; Druzdzel, 1992; Spirtes *et al.*, 1993; Pearl, 2000]. In Chapter 4, I will discuss the representation of reversible mechanisms and their role in causal reasoning.

2.4 Causal Structure, Causal Ordering, and Causal Graph

An equation-based mathematical model is qualified as a structural equation model if each equation in the model represents a conceptually distinct mechanism in a system. We may represent a static equilibrium system as a set of algebraic equations; a dynamic equilibrium system as a set of difference or differential equations; a nearly decomposable equilibrium system [Simon and Iwasaki, 1988; Simon and Rescher, 1966] as a set of mixed algebraic and difference or differential equations. Druzdzel and Simon [1993] showed that a causal Bayesian network can be represented as a structural equation model. In this dissertation, I focus on structural aspects of causal models. When performing quantitative analysis, I work with quantified causal models.

Let $\mathbf{Vars}(e)$ denotes the set of variables appearing in a structural equation e. The set of variables appearing in a set of structural equations $\mathbf{E}$ is denoted as $\mathbf{Vars}(\mathbf{E}) = \bigcup_{e \in \mathbf{E}} \mathbf{Vars}(e)$. A structural equation model can be represented as a set of structural equations $\mathbf{E} = \{e_1, e_2, \ldots, e_m\}$ over a set of variables $\mathbf{V} = \{V_1, V_2, \ldots, V_n\}$ appearing in $\mathbf{E}$, i.e., $\mathbf{V} \equiv \mathbf{Vars}(\mathbf{E})$. A variable $V_j \in \mathbf{V}$ is *exogenous* if it is determined by factors outside the model, i.e., if there exists a structural equation $e_i(V_j) = 0$ in $\mathbf{E}$. A variable is *endogenous* if it is determined by other variables in the model. I denote the sets of exogenous and endogenous variables in $\mathbf{E}$ as $\mathbf{ExVars}(\mathbf{E})$ and $\mathbf{EnVars}(\mathbf{E})$ respectively. $\mathbf{E}$ is *independent* if there does not exist an equation $e_i \in \mathbf{E}$ such that e_i is satisfied by all simultaneous solutions of any subset of $\mathbf{E} \setminus \{e_i\}$. $\mathbf{E}$ is *consistent* if the solution set of $\mathbf{E}$ is not empty. In order to ensure that $\mathbf{E}$ is independent and consistent, Simon and Rescher [1966] defined the concept *structure*.

Definition 2.3 (structure)

A structure is a set of equations $\mathbf{E}$ *where* $|\mathbf{E}| \leq |\mathbf{Vars}(\mathbf{E})|$ *such that in any subset* $\mathbf{E}' \subseteq \mathbf{E}$:

1. $|\mathbf{E}'| \leq |\mathbf{Vars}(\mathbf{E}')|$, *and*
2. *If the values of any* $|\mathbf{Vars}(\mathbf{E}')| - |\mathbf{E}'|$ *variables in* $\mathbf{Vars}(\mathbf{E}')$ *are chosen arbitrarily, then the values of the remaining* $|\mathbf{E}'|$ *variables are determined uniquely.*

The following definitions are needed for for introducing theory of causal ordering [Simon, 1953].

Definition 2.4 (self-contained structure)

A structure $\mathbf{E}$ *is self-contained if* $|\mathbf{E}| = |\mathbf{Vars}(\mathbf{E})|$.

Definition 2.5 (under-constrained structure)

A structure $\mathbf{E}$ *is under-constrained if* $|\mathbf{E}| < |\mathbf{Vars}(\mathbf{E})|$.

Definition 2.6 (minimal self-contained structure)

A self-contained structure $\mathbf{E}$ *is minimal if it does not contain any proper subset of equations in* $\mathbf{E}$ *which is self-contained.*

Definition 2.7 (strongly coupled component)

A minimal self-contained structure is a strongly coupled component if it contains more than one equation (one variable).

To complete the discussion, we say that a set of structural equations is over-constrained if the number of equations is more than the number of variables. Note that a set of over-constrained structural equations is not a structure since it violates Definition 2.3.

Definition 2.8 (over-constrained)

A set of structure equations $\mathbf{E}$ *is* over-constrained *if* $|\mathbf{E}| > |\mathbf{Vars}(\mathbf{E})|$.

To formalize the theory of causal ordering, Simon and Rescher [1966] define *structure matrix* as the qualitative representation of a set of structure equations.

Definition 2.9 (structure matrix)

A structure matrix is a qualitative representation of s set of structure equations $\mathbf{E}$*, where an element* $a_{ij} = \mathtt{x}$ *if* $V_j \in \mathbf{V}$ *participates in* $e_i \in \mathbf{E}$*; and* $a_{ij} = 0$*, otherwise.*

Example 2.1 The following SEM is the description for the match ignition: "Striking a dry match in the presence of oxygen, tinder, and fuel will cause a conflagration."[1] We define the following binary variables to describe the system: (S) struck or unstruck; (D) dry or damp; (O) oxygen or no oxygen; (I) ignited or unignited; (T) tinder or no tinder; (F) fuel or no fuel; and (C) conflagration or no conflagration. The mechanism for lighting matches is specified by the Boolean function $I = S \wedge D \wedge O$. The conflagration mechanism is specified by $C = I \wedge O \wedge T \wedge F$. The exogenous variables are S, D, O, T, and F. Figure 2.1 presents the set of structural equations in implicit function form on the left and its corresponding structure matrix on the right. □

[1] This example is first given in Simon and Rescher [1966].

$$\begin{cases} f_1(S) &= 0 \\ f_2(D) &= 0 \\ f_3(O) &= 0 \\ f_4(T) &= 0 \\ f_5(F) &= 0 \\ f_6(I,S,D,O) &= 0 \\ f_7(C,I,O,T,F) &= 0 \end{cases}$$

	S	D	O	T	F	I	C
f_1	x	0	0	0	0	0	0
f_2	0	x	0	0	0	0	0
f_3	0	0	x	0	0	0	0
f_4	0	0	0	x	0	0	0
f_5	0	0	0	0	x	0	0
f_6	x	x	x	0	0	x	0
f_7	0	0	x	x	x	x	x

Figure 2.1. The set of structural equations in implicit functional form and its corresponding structure matrix for Example 2.1.

Although each structural equation represents a causal mechanism active in a system, how does one know the causal relations among the variables? Simon [1953] proposes the *theory of causal ordering* to derive the asymmetric causal relations among variables (equations) in a self-contained structure. The theory of causal ordering only requires the qualitative knowledge, which variables appears in which structural equations, i.e, structure matrix, to derive the causal relations. The theory of causal ordering is later extended to dynamic and nearly-decomposable systems [Simon and Rescher, 1966; Simon and Iwasaki, 1988; Iwasaki and Simon, 1994]. Rather than restate the theories, I present it as *causal ordering algorithm* and explain it with examples.

The causal ordering algorithm (COA) takes a self-contained structure $\mathbf{E}$ as input and outputs a *causal graph* $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$, where $\mathbf{N}$ represents variables $\mathbf{V} = \mathbf{Vars}(\mathbf{E})$ and $\mathbf{A}$ is a set of directed arcs among $\mathbf{N}$. More precisely, $\mathbf{N}$ is a partitioning of $\mathbf{V}$, meaning that $\mathbf{N} = \{\mathbf{N}_1, \mathbf{N}_2, \ldots, \mathbf{N}_r\}$ is a pairwise disjoint sets such that $\bigcup_{i=1}^{r} \mathbf{N}_i = \mathbf{V}$, and $\mathbf{A}$ is a set of directed arcs $V_j \rightarrow \mathbf{N}_i$ where $V_j \in \mathbf{V}$, $\mathbf{N}_i \in \mathbf{N}$, and $V_j \notin \mathbf{N}_i$. The algorithm starts with *identifying* the minimal self-contained structure in $\mathbf{E}$. These identified minimal self-contained structures, $\mathbf{C}^0 = \{\mathbf{C}_1^0, \mathbf{C}_2^0, \ldots, \mathbf{C}_l^0\}$, are called *complete structures of 0-th order* and a partition $\mathbf{N}_k^0$ on $\mathbf{V}$ is created for $\mathbf{Vars}(\mathbf{C}_k^0)$, for each $\mathbf{C}_k^0 \in \mathbf{C}^0$. For each variable $V_j \in \mathbf{N}_k^0$, a corresponding node is created. When the minimal complete structure is a strongly coupled component, i.e., $|\mathbf{C}_k^0| > 1$, we draw the nodes created for variables in $\mathbf{N}_k^0$ as overlapping circles because their values need to be solved simultaneously. Next, the algorithm *removes* $\mathbf{C}^0$ from $\mathbf{E}$ for the values of $\mathbf{Vars}(\mathbf{C}^0)$ is *solved*. We denote the new structure $\mathbf{E} \setminus \mathbf{C}^0$ as $\widehat{\mathbf{E}^1}$. The algorithm then removes the columns representing $\mathbf{Vars}(\mathbf{C}^0)$ as *substituting* the solved values of $\mathbf{Vars}(\mathbf{C}^0)$ into $\widehat{\mathbf{E}^1}$ to obtain the *derived structure of the first order* $\mathbf{E}^1$. The algorithm

repeats the process of identifying, solving, removing, and substituting on the derived structure of p-th order until it is empty. In addition, whenever a partition $\mathbf{N}_k^p$ and corresponding nodes are created for a complete structure $\mathbf{C}_k^p$ in the complete structures of p-th order, the algorithm refers $\mathbf{C}_k^p$ back to its equations before any substitutions in $\mathbf{E}$, denoted as $\widehat{\mathbf{C}_k^p}$, and adds arcs from nodes representing variables in $\mathbf{Vars}(\widehat{\mathbf{C}_k^p}) \setminus \mathbf{Vars}(\mathbf{C}_k^p)$ to the nodes representing $\mathbf{N}_k^p$. Note that the causal ordering algorithm creates one-to-one mapping between equations $\widehat{\mathbf{C}_k^p}$ and variables $\mathbf{N}_k^p$, denoted as $\langle \widehat{\mathbf{C}_k^p}, \mathbf{N}_k^p \rangle$, for a self-contained causal structure. In other words, $\widehat{\mathbf{C}_k^p}$ is *mapped to* $\mathbf{N}_k^p$ and vice versa in $G(\mathbf{E})$.

Simon [1953] introduces the concept of *endogenous* and *exogenous* variables pertinent to the structure before substitutions of a complete structure of p-th order.

Definition 2.10 (endogenous and exogenous variables with respect to $\widehat{\mathbf{C}_k^p}$)
Let $\mathbf{C}^p$ and $\mathbf{C}^q$ be the complete structures of p-th and q-th order respectively in a self-contained structure $\mathbf{E}$ when applying causal ordering algorithm. Let $\widehat{\mathbf{C}_k^p}$ be the structure before any substitutions of a complete structure $\mathbf{C}_k^p \in \mathbf{C}^p$ and $V_i \in \mathbf{Vars}(\widehat{\mathbf{C}_k^p})$.

1. *V_i is endogenous with respect to $\widehat{\mathbf{C}_k^p}$, if $V_i \notin \mathbf{Vars}(\mathbf{C}^q)$ for all $q < p$.*
2. *V_i is exogenous with respect to $\widehat{\mathbf{C}_k^p}$, if $V_i \in \mathbf{Vars}(\mathbf{C}^q)$ for some $q < p$.*

The sets of endogenous and exogenous variables with respect to $\widehat{C_k^p}$ are denoted as $\mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$ and $\mathbf{ExVars}(\widehat{\mathbf{C}_k^p})$ respectively.

Simon [1953] uses the concept of endogenous and exogenous variables with respect to $\widehat{\mathbf{C}_k^p}$ to define *direct cause*.

Definition 2.11 (direct cause)
For every $\widehat{\mathbf{C}_k^p}$ in a self-contained structure $\mathbf{E}$, each $V_i \in \mathbf{ExVars}(\widehat{\mathbf{C}_k^p})$ is a direct cause of each $V_j \in \mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$.

Example 2.2 Consider applying the causal ordering algorithm to Example 2.1, COA first identifies $f_1, \ldots, f_5$ as complete subsets of 0-th order, since each equation contains one variable and can be solved for the value of this variable. Then nodes are created for each of the solved variables: S,

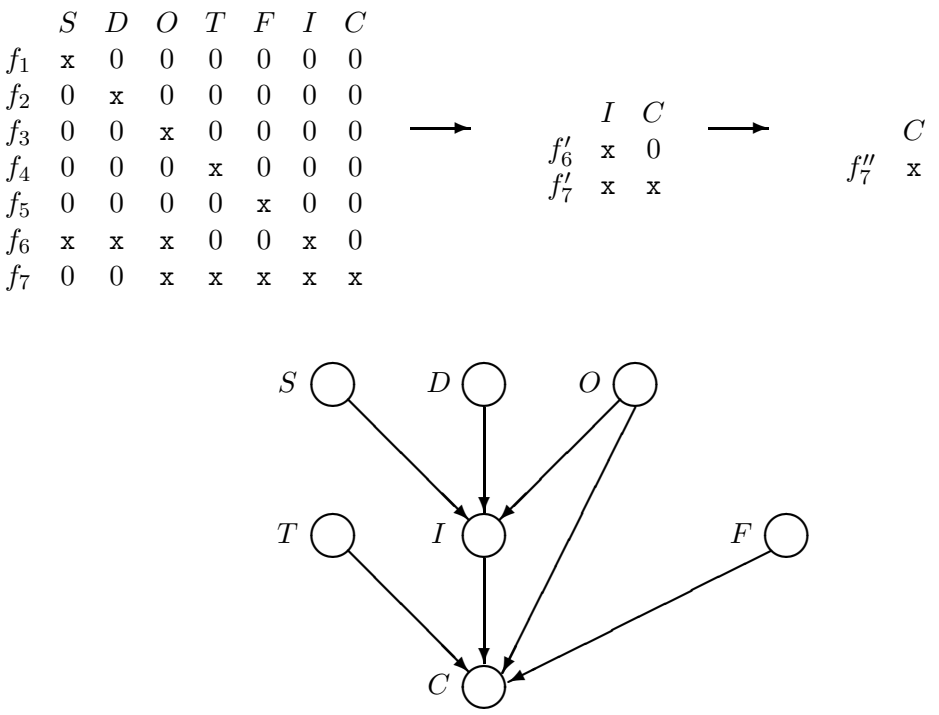


Figure 2.2. Causal ordering algorithm takes a self-contained structure as input and outputs a causal graph for Example 2.1.

D, O, T, and F. Next, the algorithm removes the solved equations $f_1, \ldots, f_5$ from the system and substitute the solved values of variables, S, D, O, T, and F, into equations f_6 and f_7, i.e, removing the columns of S, D, O, T, and F. We obtain the derived system of first order. In this derived structure, f_6' is a minimal self-contained structure, which can be solved for the values of I; correspondingly node I is created. Referring to the original equation f_6, the algorithm adds arcs from nodes O, D, S to I. And the newly derived structure of the second order only consists of f_7'' which is a self-contained structure that allows us for solving the value of C; correspondingly node C is created and the arcs from F, O, T, and I to C are added. The structure matrices for the 0-th, 1-st, and 2-nd derived structures and the causal graph generated by COA are shown in Figure 2.2. According to the causal ordering presented in the causal graph, we can read off the causal relations: (1) S, D, and O are the direct causes of I; (2) T, I, O, and F are the direct causes of C. □

Example 2.3 Consider the structure matrix presented in Figure 2.3. The causal ordering algorithm takes the structure matrix as input and identifies $\mathbf{C}^0 = \widehat{\mathbf{C}^0} = \{\{e_1\}, \{e_2\}, \{e_3\}\}$, $\widehat{\mathbf{C}^1} = \{\{e_4, e_5\}\}$, $\widehat{\mathbf{C}^2} = \{\{e_6\}, \{e_7\}\}$, and $\widehat{\mathbf{C}^3} = \{\{e_8\}\}$ to generate the causal graph. The mapping be-

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
e_1	x	0	0	0	0	0	0	0
e_2	0	x	0	0	0	0	0	0
e_3	0	0	x	0	0	0	0	0
e_4	0	x	0	x	x	0	0	0
e_5	0	x	x	x	x	0	0	0
e_6	0	0	x	0	x	x	0	0
e_7	0	0	0	x	0	0	x	0
e_8	x	0	0	0	0	0	x	x

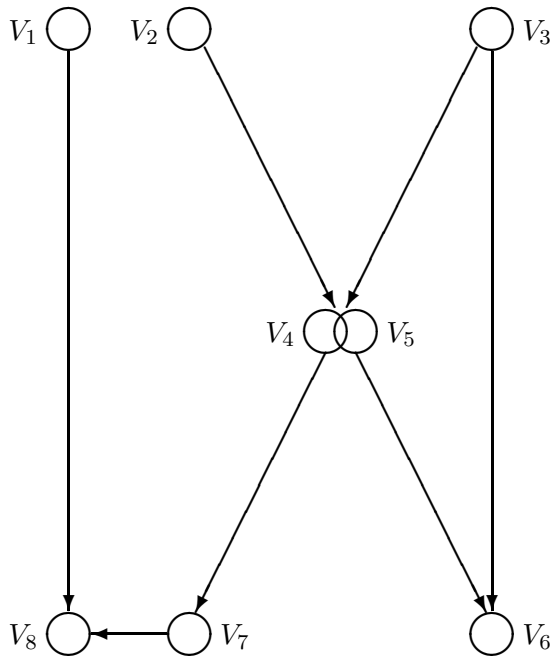


Figure 2.3. Causal ordering algorithm takes on a self-contained structure as input and outputs a causal graph with a strongly-coupled component.

tween equations and variables are $\langle e_1, V_1\rangle$, $\langle e_2, V_2\rangle$, $\langle e_3, V_3\rangle$, $\langle\{e_4, e_5\}, \{V_4, V_5\}\rangle$, $\langle e_6, V_6\rangle$, $\langle e_7, V_7\rangle$ and $\langle e_8, V_8\rangle$. From the causal graph, we can read off the causal relations among the sets of variables. We can read off direct causal relations: $\{V_4, V_5\}$ is caused by V_2 and V_3; V_6 is caused by V_3 and V_5; V_7 is caused by V_4; V_8 is caused by V_1 and V_7. We can also read off *transitive* causal relations such as V_3 is an indirect cause of V_7 because there is direct path from V_3 to V_7. However, the causal relations between V_4 and V_5 are undefined, since they are in a strongly-coupled component. Considering the endogenous and exogenous variables with respect to each complete structure, we have $\mathbf{EnVars}(\widehat{\mathbf{C}_1^0}) = \{V_1\}$, $\mathbf{EnVars}(\widehat{\mathbf{C}_2^0}) = \{V_2\}$, and $\mathbf{EnVars}(\widehat{\mathbf{C}_3^0}) = \{V_3\}$ for the complete structures of 0-th order $\widehat{\mathbf{C}}^0$. For the complete structures of 1-st order $\widehat{\mathbf{C}^1}$, we have $\mathbf{EnVars}(\widehat{\mathbf{C}^1}) = \{V_4, V_5\}$ and $\mathbf{ExVars}(\widehat{\mathbf{C}^1}) = \{V_2, V_3\}$. For $\widehat{\mathbf{C}_1^2}$, we have $\mathbf{EnVars}(\widehat{\mathbf{C}_1^2}) = \{V_6\}$ and $\mathbf{ExVars}(\widehat{\mathbf{C}_1^2}) = \{V_3, V_4, V_5\}$. For $\widehat{\mathbf{C}_2^2}$, we have $\mathbf{EnVars}(\widehat{\mathbf{C}_2^2}) = \{V_7\}$ and $\mathbf{ExVars}(\widehat{\mathbf{C}_2^2}) = \{V_4, V_5\}$. For $\widehat{\mathbf{C}^3}$, we have $\mathbf{EnVars}(\widehat{\mathbf{C}^3}) = \{V_8\}$ and $\mathbf{ExVars}(\widehat{\mathbf{C}^3}) = \{V_1, V_7\}$. □

2.5 Summary

In this chapter, I have introduced the mechanism-based view of causality and the theory of causal ordering as the causal ordering algorithm. The building blocks of causal models are variables and structural equations. Mechanisms are stable and autonomous relations among phenomena. Structural equations are mathematical representations of mechanisms. A causal model is a self-contained set of structural equations describing a system of interest. Simon [1953] developed the theory of causal ordering on a self-contained structure matrix and defined the mechanism-based view of causality by means of the causal ordering algorithm.

Chapter 3
Mechanism-based Causal Model Construction

The previous chapter describes the building blocks of causal models and the causal ordering algorithm which produces a causal graph for a self-contained causal structure to represent causal relations among variables. This chapter presents a framework for building graphical causal models based on the extended theory of causal ordering for under-constrained structures. I consider an under-constrained structure as a representation of one's intermediate understanding of a decision problem, and the process of model construction as transforming an under-constrained structure into a self-contained structure. I implement the framework as an interactive model construction module called *ImaGeNIe* in *SMILE* (Structural Modeling, Inference, and Learning Engine) and *GeNIe* (*SMILE*'s Windows user interface).

The rest of the chapter is organized as follows. Section 3.1 discusses previous approaches in supporting construction of graphical models and outlines the mechanism-based approach of causal model construction. Section 3.2 describes the *ImaGeNIe* framework. Section 3.3 discusses the representation of mechanism knowledge base. Section 3.4 presents the extension of causal ordering algorithm for under-constrained structures. Section 3.5 discusses the use of bipartite graph matching for solving the problem of causal ordering. Section 3.6 discusses the interactive modeling process. Section 3.7 presents an example of user interaction with *ImaGeNIe*.

3.1 Introduction

The quality of the advice suggested by graphical decision models depends directly on the requisiteness of the models, since the normative character of such models guarantees the correctness of the inference procedure. A model is *requisite* if it contains everything that is essential for solving the problem and no new insights about the problem will emerge by elaborating on it [Phillips, 1984]. To build a requisite model requires human intuition and creativity since the notion of requisiteness is subjective. Construction of graphical models, therefore, is laborious and demanding in terms of domain expertise. While support for obtaining model parameters,

such as prior and conditional probability distributions, has received much attention in behavioral decision theory literature (see von Winterfeldt and Edwards [1988] for a review) and in artificial intelligence [Druzdzel and van der Gaag, 2000], relatively little work has been done on composing model structure, with the work on modeling physical systems in [Iwasaki, 1988; Nayak, 1992] being notable exceptions. At the same time, there are some indications that the quality of advice is more sensitive to the model structure than to the precision of its numerical parameters [Pradhan *et al.*, 1996].

In the literature on graphical probabilistic models, such as Bayesian networks and influence diagrams, there are essentially four approaches to aid model building. The first approach focuses on providing more expressive building tools. The Noisy-OR model [Pearl, 1988; Henrion, 1989] and its generalizations [Diez, 1993; Srinivas, 1993] simplify the representation and elicitation of independence interactions among multiple causes. Heckerman [1990] developed the *similarity network* and *partition* as tools for representing *subset independence* to facilitate the structure construction and probability elicitation. The second approach, usually referred to knowledge-based model construction (KBMC), emphasizes aiding model building by automated generation of decision models from a domain knowledge-base guided by the problem description and observed information (see a special issue at the journal IEEE Transactions on Systems, Man and Cybernetics on the topic of KBMC [Breese *et al.*, 1994]). The third approach focuses on algorithms that can learn the model structure and parameters from a database of observations [Cooper and Herskovits, 1991; Pearl and Verma, 1991; Spirtes *et al.*, 1993]. Although model construction from data can reduce the knowledge engineering effort, the learning approach faces other problems such as small data sets, unmeasured variables, missing data, selection bias, and the flexibility of model granularity. Furthermore, combing several approaches is risky [Druzdzel and Díez, 2003].

While acknowledging that in the future it may be possible to build powerful computer systems that will model human creativity, sense for relevance, and simplicity, I believe that these tasks are and will long be performed better by humans. My view is that model building, a task that relies on all these capacities, is best implemented as an interactive process. The fourth approach on aiding model construction that is most related to my work is to apply system engineering and knowledge engineering techniques for aiding the process of building Bayesian networks. Laskey and Mahoney [1996; 1997] address the issues of modularization, object-orientation, knowledge-base,

and evaluation in a spiral model of development cycle. Koller and Pfeffer [1997; 1999] developed Object-Oriented Bayesian Networks (OOBN) that use objects as organizational units to reduce the complexity of modeling and increase the speed of inference.

My approach on aiding model construction is based on the mechanism-based view of causality, where mechanisms are building blocks of a model and the theory of causal ordering determines the causal structure of a model. As they encode our understanding of local interactions among variables, mechanisms are fairly model independent and can be easily reused in various models. Similarly to the abstraction of object-hierarchy, mechanisms can be organized hierarchically in nearly decomposable system [Iwasaki and Simon, 1994]. In addition, mechanism-based view of causality provides a valuable heuristic for acquiring and managing causal knowledge.

In my framework, I encode mechanisms as functional relations among variables and, wherever causal mechanisms are asymmetric, the direction of causal influence among variables. I extend Simon's causal ordering algorithm [Simon, 1953] to develop a modeling process that uses the output graph of this algorithm in the interaction with users. I assist the model building process by helping users (1) to identify a set of mechanisms related to the current model and to bring them into model workspace (2) to integrate the newly added mechanisms with the model under construction (3) to specify the manipulated variables, and (4) to extract reusable mechanisms from existing models into the knowledge base. The final model structures generated by my modeling process are guaranteed to be causal if the underlying structural equations reflect causal mechanisms of the modeled problem.

One important aspect of causal reasoning is to allow users to predict the effect of manipulation, i.e., changes in structure. The users of causal models (and that includes autonomous robots) can ask questions like "What will happen if I perform action A?" Manipulation is especially important in strategic planning, where it is important to derive creative decision options and not only to evaluate existing decision options. Furthermore, a user may want to explore the possibility of manipulating different variables in the process of creating a model. Supporting this manipulation is not straightforward, as some mechanisms may be reversible, i.e., acting in reverse direction. For example, when driving up the hill, car engine causes the wheels to turn; but when driving down the hill in a low gear, the model should be able to predict that the wheels will cause the engine to slow down. My approach supports causal modeling that includes reversible mechanisms and offers

an integrated framework for building and using causal models. I will present the theoretical and algorithmic aspects on supporting changes in structure in Chapter 4.

3.2 The Framework

I develop an interactive and iterative model construction environment, called *ImaGeNIe*, that assists users in building causal models. I use the causal ordering algorithm to generate the network structure of a causal model, which can later be associated with different node types and parameters. Figure 3.1 shows the architecture of *ImaGeNIe*. It includes three knowledge structures: *mechanism knowledge bases*, which holds domain knowledge expressed as causal mechanisms, *model building workspace*, which serves as a blackboard for model composition, and *models*. The domain knowledge can be maintained either by the *equation authoring interface*, where model builders can compose structural equations directly, or by the *mechanism extraction operation* that enables model builders to extract reusable causal mechanisms from existing models. Model builders can use *hierarchy navigation interface* to locate the mechanisms of interest and select them into the model building workspace with assistance of the *mechanism selection operation*. In addition to mechanism selection and traditional model authoring operations, model builders can *manipulate variables* and *merge mechanisms* as the model building process evolves. The underlying *causal ordering* module will restructure the models according to the users' interactions with the system.

3.3 Mechanism Knowledge Base

In *ImaGeNIe*, the mechanism knowledge base is organized as a hierarchical system that consists of subsystems and causal mechanisms as its fundamental building elements. The hierarchical approach not only assists domain experts to express their domain knowledge in cognitively meaningful units but also helps knowledge engineers to access stored mechanisms easily. The approach is similar to type-hierarchy in [Koller and Pfeffer, 1997; Laskey and Mahoney, 1997] but without imposing the inheritance constraint since knowledge can be possibly organized hierarchically from different perspectives. Appendix A describes the XML schema of the mechanism knowledge base in *ImaGeNIe*.

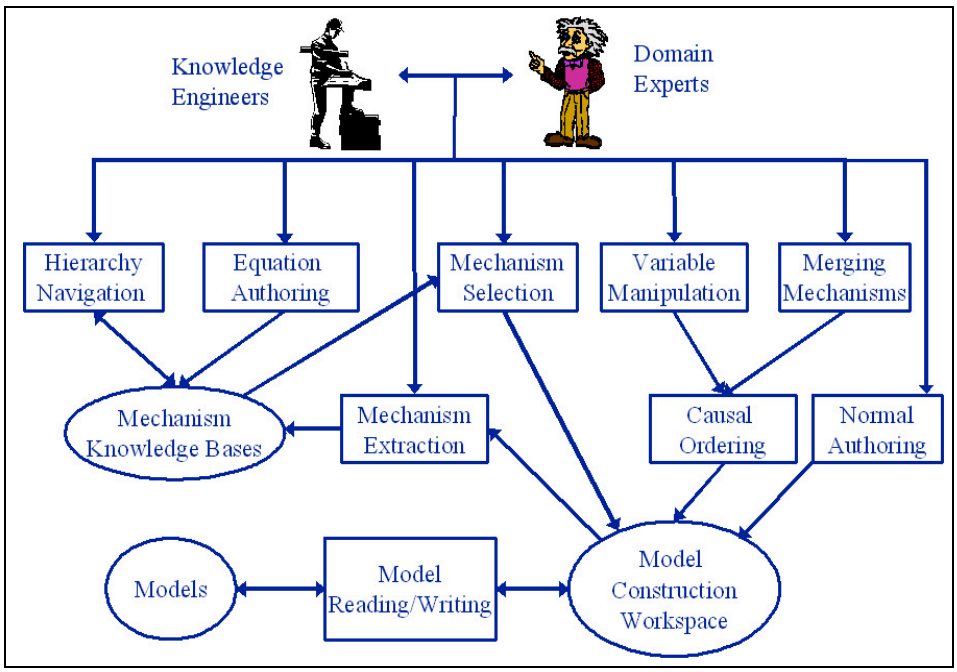


Figure 3.1. Interactive and iterative model construction system architecture. The arcs show the direction of the information flow.

The fundamental knowledge units in mechanism knowledge base are causal mechanisms which are represented as structural equations. Users can specify structural equations in implicit functional forms or explicit functional forms such as algebraic functions, conditional probability tables, truth tables, value/utility tables, and choice tables. While most mechanisms will be described in one, perhaps their only, mode of operation, some mechanisms will be described in different modes of operation because when those mechanisms are embedded in different operation contexts, the causal relations among variables may be reversed. The reversibility of a mechanism will be formally introduced in Chapter 4.

To aid the process of model building, I define the *manipulability* and *observability* properties for each variable in the domain knowledge base. On the manipulability property, a variable can be *manipulable* or *non-manipulable*. A variable is manipulable if it can be manipulated directly, i.e., the value of a manipulable variable can be set directly, by forces outside a model in a modeling domain. A variable is non-manipulable if the value of the variable has to be derived from a model in a modeling domain. For example, the mechanism describing the degree of sunshine (S) and my

behavior of wearing sunglasses (SG) can be represented by a structural equation $f(S, SG) = 0$. I define both S and SG as manipulable variables in the modeling domain. The degree of sunshine S is determined by the external force of nature. Wearing sunglasses or not (SG) could be controlled by me. Note that the manipulability property of a variable in a modeling domain is different from its appearance as an exogenous variable in a model. For example, I can have $g(S) = 0$, which describes the degree of sunshine, together with $f(S, SG) = 0$ to model the situation where I wear my sunglasses because of the sunshine ($S \rightarrow SG$). In this example, I have S as an exogenous variable, but not SG. However, SG is still manipulable in my modeling domain, since I can always wear sunglasses regardless of the degree of sunshine. In such case, I have the model with structural equations $g(S) = 0$ and $h(SG) = 0$ (representing the behavior of wearing sunglasses regardless of sunshine) with a causal graph in which SG is disconnected from S and both are exogenous variables with respect to the model. In other words, a manipulable variable is not necessary manipulated in a model, but an exogenous variable is manipulated in a model and it has to be a manipulable variable in the modeling domain. I call the unmanipulated manipulable variables in a model as *potential policy variables*. On the observability property, a variable can be *observable* or *unobservable*. It is sometime desirable to associate other properties with variables to facilitate the use of models. For example, one may want to associate the properties such as *manipulation cost*/*observation cost* with manipulable/observable variables to incorporate the modeling of costs into causal models.

3.4 Extended Theory of Causal Ordering

In *ImaGeNIe*, we the model construction process is considered as a reflection of our problem solving. The under-constrained structures emerged in such process reveal different stages of problem solving. Mechanisms in different under-constrained structures are structural relations recognized by modelers as pertinent to each stage of problem solving. Exogenous variables in under-constrained structures are outside influences that have been committed by modelers. An under-constrained model cannot be drawn as a directed acyclic graph, as the causal interactions are not completely determined until a model is self-contained. However, it is desired to have a graphical representation of under-constrained models during the whole process of model construction, since the graphical representation can help modelers identify focus and commitments of the outside influences. I extend

Simon's causal ordering algorithm to explicate the causal ordering that has been identified in under-constrained models. In addition, I propose a graphical representation to depict the causal ordering in an informative graphical form that aims to help users in model building.

In order to formalize the extended theory of causal ordering, I restate the theorem that was originally proven by Simon [1953].

Theorem 3.1 (disjunct property)

Let $\mathbf{A}$ *and* $\mathbf{B}$ *be two minimal self-contained subsets of a structure* $\mathbf{E}$*. Then the structural equations of* $\mathbf{A}$ *and* $\mathbf{B}$*, and likewise the variables in* $\mathbf{A}$ *and* $\mathbf{B}$ *are disjunct.*

Based on Theorem 3.1, I prove the following lemma.

Lemma 3.1

Let $\mathbf{E}$ *be a structure and* $\mathbf{E}'$ *be the derived set of structural equations from* $\mathbf{E}$ *by applying identification, solving, removing, and substitution. If* $\mathbf{E}'$ *is not empty, then* $\mathbf{E}'$ *is a structure.*

Proof: In Section 2.4, I describe the operations of identification, solving, removing, and substitution on a self-contained structure. Now, I apply these operations with the same definitions to a structure $\mathbf{E}$. In the process of identification, let $\mathbf{C}$ be the union of all the minimal self-contained subsets, i.e., $\mathbf{C} = \mathbf{C}^1 \cup \mathbf{C}^2 \cup \ldots \cup \mathbf{C}^k$, and the remainder $\mathbf{R}$. We know that $\mathbf{R}$ is not empty since $\mathbf{E}'$ is not empty as given in the premise. Suppose that $\mathbf{E}'$ is not a structure. Then there exists a subset $\mathbf{E}^*$ of $\mathbf{E}'$ such that $|\mathbf{E}^*| > \mathbf{Vars}(\mathbf{E}^*)$. Let $\mathbf{R}^*$ be the subset of $\mathbf{R}$ where $\mathbf{E}^*$ derives from. We know that the number of equations in $\mathbf{R}^*$ is the same as the number of equations in $\mathbf{E}^*$, i.e, $|\mathbf{R}^*| = |\mathbf{E}^*|$. Now, consider the subset $\mathbf{F} = \mathbf{C} \cup \mathbf{R}^*$. The equations of $\mathbf{C}$ and $\mathbf{R}^*$ are disjunct because $\mathbf{C}$ and $\mathbf{R}$ are disjunct and $\mathbf{R}^* \subseteq \mathbf{R}$. Therefore, $|\mathbf{F}| = |\mathbf{C}| + |\mathbf{R}^*| = |\mathbf{C}| + |\mathbf{E}^*|$. Since $\mathbf{E}^*$ derives from $\mathbf{R}^*$ by substitution, the variables appearing in $\mathbf{R}^*$ are either in $\mathbf{C}$ or in $\mathbf{E}^*$. Consequently, the variables in $\mathbf{F}$ are either in $\mathbf{C}$ or in $\mathbf{E}^*$. Moreover, the variables in $\mathbf{C}$ and $\mathbf{E}^*$ are disjunct because $\mathbf{E}^*$ derives from $\mathbf{R}^*$ by substituting out the variables in $\mathbf{C}$. Therefore, $|\mathbf{Vars}(\mathbf{F})| = |\mathbf{Vars}(\mathbf{C})| + |\mathbf{Vars}(\mathbf{E}^*)|$. Since the equations of $\mathbf{C}^i$, and likewise the variables in $\mathbf{C}^i$, are disjunct by Theorem 3.1, we have $|\mathbf{Vars}(\mathbf{C})| = \sum_i |\mathbf{Vars}(\mathbf{C}^i)|$ and $|\mathbf{C}| = \sum_i |\mathbf{C}^i|$. Hence $|\mathbf{Vars}(\mathbf{C})| = |\mathbf{C}|$. Therefore, $|\mathbf{F}| = |\mathbf{C}| + |\mathbf{R}^*| = |\mathbf{C}| + |\mathbf{E}^*| > |\mathbf{C}| + |\mathbf{Vars}(\mathbf{E}^*)| = |\mathbf{Vars}(\mathbf{C})| + |\mathbf{Vars}(\mathbf{E}^*)| = |\mathbf{Vars}(\mathbf{F})|$, i.e., the number of equations of $\mathbf{F}$ is greater than the number of variables of $\mathbf{F}$. In other words,

the set $\mathbf{F}$ violates Definition 2.3 contradicting the fact that $\mathbf{E}$ is a structure. We conclude that $\mathbf{E}'$ must be a structure. □

Given Lemma 3.1, I can keep applying identification, solving, removing, and substitution operations on derived structure till either $\mathbf{E}'$ is empty or there are no more minimal self-contained structures that can be identified. If $\mathbf{E}'$ is empty, I know that $\mathbf{E}$ is self-contained. If $\mathbf{E}'$ is not empty and no more self-contained structures can be identified, I know that $\mathbf{E}$ is under-constrained and I call the final $\mathbf{E}'$ the *derived strictly under-constrained structure*.

Definition 3.1 (strictly under-constrained structure)

An under-constrained structure is strictly under-constrained if it does not contain any self-contained structures.

Theorem 3.2

A structure $\mathbf{E}$ *is under-constrained if and only if there exists a derived strictly under-constrained structure in* $\mathbf{E}$.

Proof: ($\Rightarrow$) By Definition 2.5, we know that $|\mathbf{Vars}(\mathbf{E})| > |\mathbf{E}|$. Applying Lemma 3.1 on $\mathbf{E}$, we derive a strictly under-constrained subsets $\mathbf{E}'$ that has $|\mathbf{Vars}(\mathbf{E}')| > |\mathbf{E}'|$.

($\Leftarrow$) Since a derived strictly under-constrained subset must be obtained by the process of identification, solving, and substitution operations, therefore, by definition $\mathbf{E}$ is a structure, i.e., $\mathbf{E}$ can be either under-constrained or self-contained. If $\mathbf{E}$ is self-contained, then the derived structure $\mathbf{E}'$ must be empty; otherwise, we can keep applying the process of identification, solving, and substitution operations. We conclude that we cannot obtain a derived strictly under-constrained structure from a self-contained $\mathbf{E}$. Therefore, $\mathbf{E}$ must be under-constrained. □

Based on Theorem 3.2, I present the extended causal ordering algorithm in Figure 3.2. The input of the algorithm is a structure matrix $\mathbf{E}$. The output is a graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ where $\mathbf{N}$ represents $\mathbf{Vars}(\mathbf{E})$, $\mathbf{A}$ is a set of directed arcs, and $\mathbb{A}$ is a set of undirected arcs among $\mathbf{N}$. The algorithm essentially follows the steps of identification, solving, removing, and substitution as Simon's causal ordering algorithm until there are no more self-contained subsets that can be identified from the derived structure. The algorithm will explicitly depicts the causal relations and relevant relations encoded in the strictly under-constrained subset, if there remains one.

Procedure *ExtendedCausalOrdering*($\mathbf{E}$)

Input: A structure matrix $\mathbf{E}$.

Output: A graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ where $\mathbf{N}$ represents $\mathbf{Vars}(\mathbf{E})$, $\mathbf{A}$ is a set of directed arcs, and $\mathbb{A}$ is a set of undirected arcs among $\mathbf{N}$.

1. $i := 0$; $\mathbf{E}^i := \mathbf{E}$.
2. **while** ($\exists$ $\mathbf{C}^i \subseteq \mathbf{E}^i$)
3. **for each** $\mathbf{C}^i_k \in \mathbf{C}^i$
4. Create a node N_j for each $V_j \in \mathbf{Vars}(\mathbf{C}^i_k)$.
5. Add arcs from each N_l representing $V_l \in \mathbf{ExVars}(\widehat{\mathbf{C}^i_k})$ to N_j.
6. **if** $|\mathbf{Vars}(\mathbf{C}^i_k)| > 1$ **then**
7. Overlap nodes representing $\mathbf{Vars}(\mathbf{C}^i_k)$.
8. **end if**
9. **end for each**
10. $\widehat{\mathbf{E}^i} := \mathbf{E}^i \setminus \mathbf{C}^i$.
11. Remove $\mathbf{Vars}(\mathbf{C}^i)$ from $\widehat{\mathbf{E}^i}$ to derive $\mathbf{E}^{i+1}$.
12. $i := i + 1$.
13. **end while**
14. **if** $\mathbf{E}^i \neq \emptyset$
15. **for each** $e \in \mathbf{E}^i$
16. Create nodes $\mathbf{N}_e$ representing $\mathbf{Vars}(e)$.
17. Add arcs from nodes representing $\mathbf{ExVars}(\widehat{e})$ to $\mathbf{N}_e$.
18. **end for each**
19. Create pair-wise undirected arcs among $\mathbf{N}_{\mathbf{E}^i}$.
20. **end if**

Figure 3.2. Extended causal ordering algorithm.

The graph generated by the extended causal ordering algorithm is designed to aid the process of model construction. Unlike the original causal ordering algorithm, each variable in the structure is represented as a separate node so that the modeler can access and manipulate on it directly. The

set of directed arcs depicts causal relations encoded in an under-constrained structure. A strongly-coupled component is drawn as a set of overlapped nodes. The set of undirected arcs depict relevant relations among undetermined variables. Undirected arcs present relevant but undetermined causal relations among variables so that model builder can focus on clarifying the mechanisms governing these variables to make a model self-contained.

3.5 Bipartite Graph Matching and Causal Ordering

The extended causal ordering algorithm outlined in Figure 3.2 and the causal ordering algorithm discussed in Section 2.4 are in worst-case exponential time algorithms. Nayak [1994] discussed a polynomial time algorithm, based on the bipartite graph matching, for causal ordering over self-contained structures. Although Nayak's work focused on the automated modeling of physical systems, he did not address the problem of causal ordering for under-constrained structures in model construction. Nayak's work on causal ordering was based on the work of Serrano and Gossard [1987] where they focused on the constraint management in conceptual design. Given a set of equations, they recognized that the complete matching between the set of equations and the set of variables yields the causal ordering. Although Serrano and Gossard discussed the use of bipartite graph matching for detecting under-constrained and over-constrained systems, they did not discuss causal relation among variables in these systems.

In Section 3.5.1, I summarize the link of causal ordering and the bipartite graph matching in [Serrano and Gossard, 1987; Nayak, 1994] and demonstrate how to use the bipartite graph matching to derive causal graphs. Section 3.5.2 derives the method that uses the bipartite graph matching in the extended causal ordering for under-constrained structures.

3.5.1 Complete Matching and Causal Ordering

Given a self-contained structure $\mathbf{E}$, the causal ordering algorithm creates one-to-one mappings $\langle \widehat{\mathbf{C}_k^p}, \mathbf{EnVars}(\widehat{\mathbf{C}_k^p}) \rangle$, between the equations of minimal self-contained structures, $\widehat{\mathbf{C}_k^p}$, and its endogenous variables, $\mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$ (See Section 2.4). The concept of direct cause, defined in Definition 2.11, states that for each $\widehat{\mathbf{C}_k^p}$ in a self-contained structure $\mathbf{E}$, $\mathbf{ExVars}(\widehat{\mathbf{C}_k^p})$ are direct causes of $\mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$. The bottleneck of the causal ordering algorithm discussed in Section 2.4 is in the

step of identifying these one-to-one mappings. Serrano and Gossard [1987] suggested the use of the bipartite graph matching in finding these one-to-one mappings. The bipartite graph representation of structure **E** can be formalized as follows.

Definition 3.2 (bipartite graph of a structure E)
The bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$ *of a structure* **E** *consists of two disjoint sets of nodes* $\mathbf{N_E}$ *and* $\mathbf{N_V}$*, and a set of undirected arcs* $\mathbb{A}$*, where*

1. *for each equation* $e_i \in \mathbf{E}$*, there is a node* $N_{e_i} \in \mathbf{N_E}$*.*
2. *for each variable* $V_i \in \mathbf{V}$*, where* $\mathbf{V} = \mathbf{Vars}(\mathbf{E})$*, there is a node* $N_{V_i} \in \mathbf{N_V}$*, and*
3. *for each equation* $e_i \in \mathbf{E}$*, there is a set of undirected arcs* (N_{e_i}, N_{V_j}) *in* $\mathbb{A}$ *for each* $V_j \in \mathbf{Vars}(e_i)$*.*

A *matching* in a bipartite graph is a set of arcs such that no two nodes in the matching share an arc. A node N_{e_i} is *matched* to a node N_{V_j} (or vise versa) with respect to a matching $\mathbb{A}_m \subseteq \mathbb{A}$, if the arc $(N_{e_i}, N_{V_j}) \in \mathbb{A}_m$. A matching is *complete* if and only if each node in the graph is covered by an arc in the matching [Even, 1979]. Serrano and Gossard [1987] and Nayak [1994] recognized that a complete matching of $BG(\mathbf{E})$ resembles one-to-one mapping in the causal ordering of **E**. Given a self-contained structure **E**, I formalize the algorithm, denoted as COA_{BGM}, that uses the bipartite graph matching to generate the causal graph of **E** as in Figure 3.3.

COA_{BGM} takes a self-contained structure **E** as input and outputs a causal graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$. It first constructs the bipartite graph $BG(\mathbf{E})$ and then identifies a complete matching $\mathbb{A}_m$ of $BG(\mathbf{E})$. Each arc (N_{e_i}, N_{V_j}) in $\mathbb{A}_m$ is interpreted as a direct dependency relation where $\mathbf{Vars}(e_i) \setminus V_j$ directly determine V_j with equation e_i. Based on this interpretation, a directed graph $DG(\mathbf{E})$ is created to depict the dependency relations encoded in **E** by $\mathbb{A}_m$. The algorithm then identifies the strongly-connected components in $DG(\mathbf{E})$ and depicts those that have more thane one element in the component as strongly-coupled components. Interpreting a strongly-connected component with more than one element as a strongly-coupled component is because variables within such strongly-connected component are interdependent in $DG(\mathbf{E})$. The algorithm finally outputs the modified acyclic graph $DG(\mathbf{E})$ as $G(\mathbf{E})$.

Procedure $COA_{BGM}(\mathbf{E})$

Input: A self-contained structure matrix $\mathbf{E}$.

Output: A causal graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$, where $\mathbf{N}$ represents $\mathbf{Vars}(\mathbf{E})$ and $\mathbf{A}$ is a set of directed arcs depicting causal relations among $\mathbf{N}$.

1. Create the bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$.
2. Identify a complete matching $\mathbb{A}_m \subset \mathbb{A}$ of $BG(\mathbf{E})$.
3. Construct a directed graph $DG(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$ with respect to $\mathbb{A}_m$, where
4. the set of nodes $\mathbf{N}$ corresponds to $\mathbf{N_V}$, and
5. **for each** $(N_{e_i}, N_{V_j}) \in \mathbb{A}_m$
6. there exists in $\mathbf{A}$ a set of directed arcs, $N_k \rightarrow N_j$, where N_k represents
7. each $V_k \in \{\mathbf{Vars}(e_i) \setminus \{V_j\}\}$ and N_j represents V_j.
8. **end for each**
9. Identify the strongly-connected components $\mathbf{S}$ in $DG(\mathbf{E})$.
10. **for each** strongly-connected component $S_i \in \mathbf{S}$
11. **if** $|S_i| > 1$
12. Collect the set of parents of each $N_j \in S_i$ as $\mathbf{P}$.
13. Remove all incoming arcs of each $N_j \in S_i$.
14. Create a partition $\mathbf{N}_i$ with all nodes in S_i.
15. Draw $N_j \in \mathbf{N}_i$ as overlapped for the strongly-coupled component S_i.
16. Create directed arcs $P_i \rightarrow \mathbf{N}_i$ for all $P_i \in \mathbf{P}$.
17. **end if**
18. **end for each**
19. Return $DG(\mathbf{E})$ as $G(\mathbf{E})$.

Figure 3.3. Causal ordering algorithm based on bipartite graph matching.

COA_{BGM} is in the worst-case polynomial in time. In Figure 3.3, the graph construction for both $BG(\mathbf{E})$ and $DG(\mathbf{E})$, listed in Line 1 and Line 3–8 respectively, can be done in linear time. Similarly, the graph manipulation in Lines 10–18 and the identification of strongly-connected components in Line 9 can also be done in linear time. For the bipartite graph matching in Line 2, there are well-

known polynomial time algorithms. For example, Papadimitriou and Stelglitz [1982] described a bipartite graph matching algorithm using network-flow technique with time complexity $O(n^{1/2}a)$, where $n = |\mathbf{N_E}| = |\mathbf{N_V}|$ and $a = |\mathbb{A}|$. Therefore, COA_{BGM} is a worst-case polynomial time algorithm.

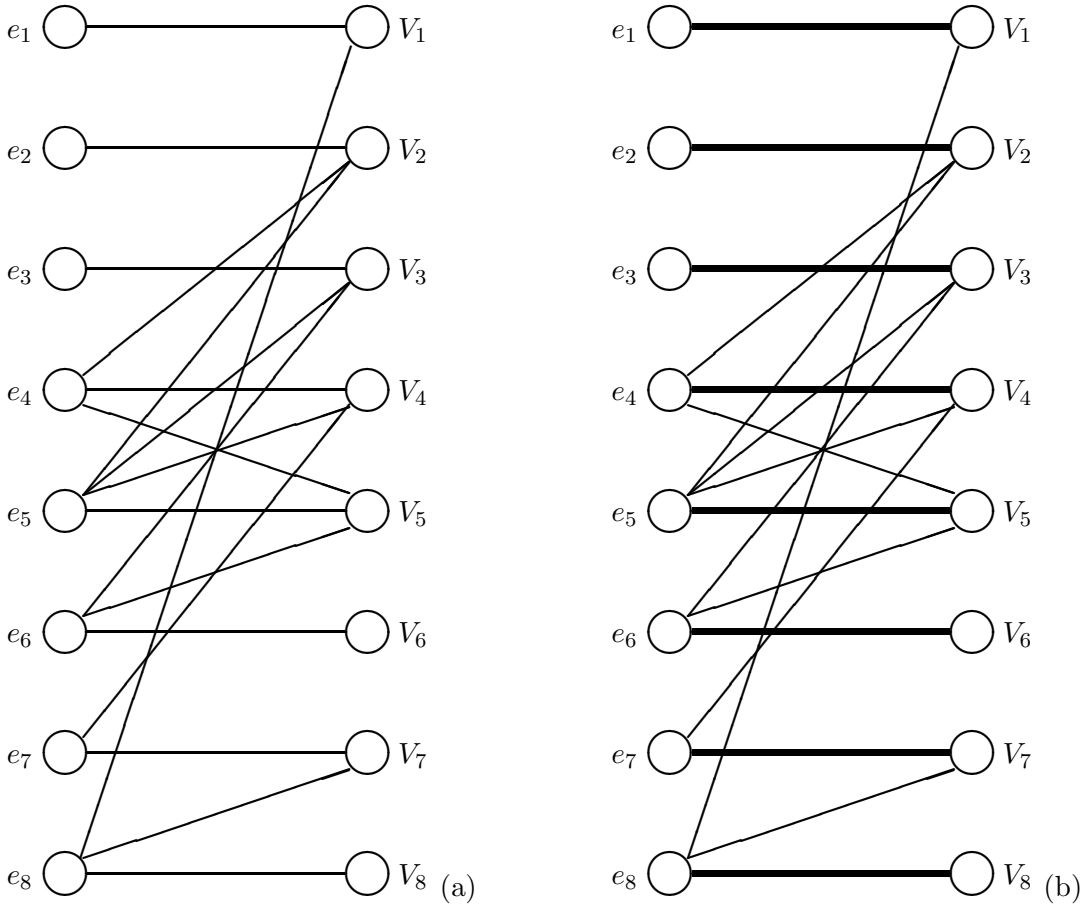


Figure 3.4. The bipartite graph representation of the self-contained structure in Example 2.3 is shown in (a). A complete matching of the bipartite graph (a) is depicted by bold arcs in (b).

Example 3.1 Consider applying the COA_{BGM} algorithm on the structure in Example 2.3. COA_{BGM} first creates the corresponding bipartite graph of the structure as shown in Figure 3.4 (a). Next, COA_{BGM} computes the complete matching as shown in Figure 3.4 (b), where each bold arc depicts an arc in the matching. Third, COA_{BGM} creates a directed graph, shown in Figure 3.5 (a), according to the complete matching. Finally, COA_{BGM} generates the causal graph by identifying the strongly-coupled components and creating corresponding partitions with modifications over the set

of incoming arcs of strongly-coupled components (See Figure 3.5 (b)). □

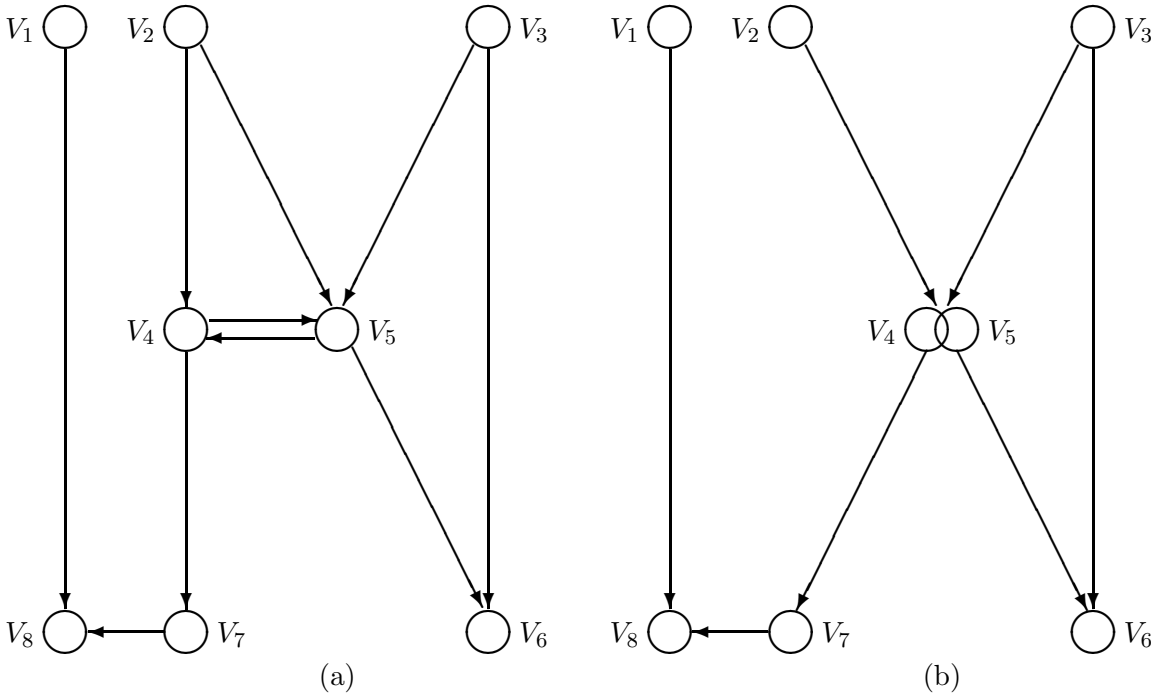


Figure 3.5. (a): COA_{BGM} creates the directed graph according to the complete matching in Figure 3.4 (b). (b): COA_{BGM} creates the causal graph by identifying the strongly-connected component $\{V_4, V_5\}$ in (a) and grouping it into a strongly-coupled component.

Serrano and Gossard [1987] observed that there may exist more than one complete matching in Line 2 of COA_{BGM}, resulting from the strongly-coupled components for a given self-contained structure. For example, an alternative complete matching for the example in Figure 3.4 would have been (e_5, V_4) and (e_4, V_5) instead of (e_4, V_4) and (e_5, V_5). Nayak [1994] proved that there exists an unique *transitive closure* for all possible $DG(\mathbf{E})$s, created in Lines 3–8 of COA_{BGM}, with respect to all possible complete matchings for a self-contained structure $\mathbf{E}$. This theorem allows us to transform one $DG(\mathbf{E})$ for a complete matching to the causal graph $G(\mathbf{E})$ that represents the unique transitive closure over all possible $DG(\mathbf{E})$s. Nonetheless, both algorithms in [Nayak, 1994; Serrano and Gossard, 1987] stop at Line 8 of COA_{BGM} and with additional discussion for Line 9, since Serrano and Gossard focused on the evaluation of constraints and Nayak focused on generating causal explanations for physical systems, where a succinct graphical representation for causal ordering was not the focus.

One important difference between graphs generated by the *COA* in [Simon, 1953] and graphs generated by procedure in [Nayak, 1994] is the depiction of the causal relations among variables in a strongly-coupled component. Variables in a strongly-coupled component are interdependent, but no causal relations among them are explicitly identified by *COA*. On the other hand, graphs generated by the procedure of Nayak hypothesize the causal relations among variables in strongly-coupled components. For example, Figure 3.5 (a) shows that the direct causes of V_4 are $\{V_2, V_5\}$ and the direct causes of V_5 are $\{V_2, V_3, V_4\}$, but Figure 3.5 (b) shows that the direct causes of $\{V_4, V_5\}$ are $\{V_2, V_3\}$ and variables V_4 and V_5 are interdependent. The causal relations among variables in strongly-coupled components was the subject of a debate between [Iwasaki and Simon, 1986a; 1986b] and [de Kleer and Brown, 1986]. I take the view of Simon and treat variables in a strongly-coupled component as interdependent.

3.5.2 Maximum Matching and Extended Causal Ordering

In addition to the relation between complete matching and causal ordering, Serrano and Gossard [1987] discussed the use of *maximum matching* in identifying under-constrained and over-constrained systems. Since the focus of their work was constraint management, they did not discuss causal ordering in under-constrained systems. In this section, I develop a method that derives the extended causal ordering in polynomial time using maximum matching.

A maximum matching in a bipartite graph is a matching with maximum cardinality. A complete matching is by definition a maximum matching. The following theorem states the relation between the bipartite graph matching and a structure $\mathbf{E}$.[1]

Theorem 3.3

Let $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$ *be the bipartite graph of a structure* $\mathbf{E}$*. Then* $\mathbf{E}$ *is self-contained, if there is a complete matching in* $BG(\mathbf{E})$*; and under-constrained, otherwise.*

Proof: According to Hall's theorem ([Even, 1979, p.p. 137-138]), a bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$ has complete matching if and only if (a) $|\mathbf{N_E}| = |\mathbf{N_V}|$ and (b) for every subset $\mathbf{N_{E'}} \subseteq \mathbf{N_E}$, $|\mathbf{N_{E'}}| \leq |\mathbf{N_{V'}}|$, where $\mathbf{N_{V'}}$ are nodes connected with $\mathbf{N_{E'}}$ in $\mathbb{A}$. The $BG(\mathbf{E})$ of a self-contained structure $\mathbf{E}$ satisfies criteria (a) and (b) by Definition 2.3 and 2.4. Therefore, if there is

[1] Note that Theorem 3.3 also provides the foundation for Line 2 of COA_{BGM}.

a complete matching in $BG(\mathbf{E})$ of a structure $\mathbf{E}$, then $\mathbf{E}$ is self-contained. The $BG(\mathbf{E})$ of an under-constrained structure $\mathbf{E}$, on the other hand, violates the criteria (a) by Definition 2.5. Therefore, the $BG(\mathbf{E})$ of an under-constrained structure $\mathbf{E}$ cannot have a complete matching according to Hall's theorem. □

Given Theorem 3.3 and the fact that a complete matching is also a maximum matching, I can apply an algorithm finding maximum matching to the $BG(\mathbf{E})$ of a given structure $\mathbf{E}$ and check whether a complete matching is found. If yes, I have a self-contained structure and a complete matching that can lead me to construct the causal graph of the self-contained structure $\mathbf{E}$. If not, I have an under-constrained structure with a maximum matching. Next, I show how to derive the extended causal ordering using the maximum matching of a bipartite graph of an under-constrained structure.

For any matching $\mathbb{A}_m$ of a bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$ of a structure $\mathbf{E}$, I denote the set of matched nodes in $\mathbb{A}_m$ as $\mathbf{N}_{\mathbf{E}_m} \subseteq \mathbf{N_E}$ and $\mathbf{N}_{\mathbf{V}_m} \subseteq \mathbf{N_V}$, and the set of unmatched nodes as $\mathbf{N}_{\mathbf{E}_{\overline{m}}} \subseteq \mathbf{N_E}$ and $\mathbf{N}_{\mathbf{V}_{\overline{m}}} \subseteq \mathbf{N_V}$. I prove the following lemma to show that given any under-constrained structure $\mathbf{E}$, all $\mathbf{N_E}$ are matched in any maximum matching $\mathbb{A}_m$ of $BG(\mathbf{E})$, i.e., $\mathbf{N}_{\mathbf{E}_m} = \mathbf{N_E}$ and $\mathbf{N}_{\mathbf{E}_{\overline{m}}} = \emptyset$.

Lemma 3.2

Any maximum matching $\mathbb{A}_m$ of the bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A} \rangle$ of an under-constrained structure $\mathbf{E}$ has all $\mathbf{N_E}$ matched and some $\mathbf{N}_{\mathbf{V}_{\overline{m}}} \subset \mathbf{N_V}$ unmatched.

Proof: I prove this lemma by contradiction. Assume that $\mathbb{A}_m$ is a maximum matching of $BG(\mathbf{E})$ with $\mathbf{N}_{\mathbf{E}_{\overline{m}}} \neq \emptyset$. Since $\mathbf{E}$ is an under-constrained structure and $|\mathbf{N}_{\mathbf{E}_m}| = |\mathbf{N}_{\mathbf{V}_m}|$, we have $|\mathbf{N}_{\mathbf{E}_{\overline{m}}}| < |\mathbf{N}_{\mathbf{V}_{\overline{m}}}|$ and $\mathbf{N}_{\mathbf{V}_{\overline{m}}} \neq \emptyset$.

To simplify our discussion, we focus on an unmatched node $N \in \mathbf{N}_{\mathbf{E}_{\overline{m}}}$. We know that N is only connected with nodes in $\mathbf{N}_{\mathbf{V}_m}$, but not with nodes in $\mathbf{N}_{\mathbf{V}_{\overline{m}}}$. Otherwise, it contradicts that $\mathbb{A}_m$ is a maximum matching, since we can increase the cardinality of $\mathbb{A}_m$ by including the arc that connects N and the node in $\mathbf{N}_{\mathbf{V}_{\overline{m}}}$.

Let $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{V}_m}$ be the set of nodes that connected with N and $\mathbf{Y} \subseteq \mathbf{N}_{\mathbf{E}_m}$ be the set of nodes that matched with $\mathbf{X}$ in $\mathbb{A}_m$. We know that $\mathbf{Y}$ must connect with nodes other than $\mathbf{X}$ in $\mathbb{A}$; otherwise, $|\mathbf{Y} \cup N| > |\mathbf{X}|$, contradicting with the bipartite graph representation of a structure. If any $Y \in \mathbf{Y}$ is

connected with a node $Z \in \mathbf{N}_{\mathbf{V}_{\overline{m}}}$, we can replace the matched arc (Y, X) in $\mathbb{A}_m$, where $X \in \mathbf{X}$ and X is connected with N, by (N, X) and (Y, Z), which contradicts the fact that $\mathbb{A}_m$ is a maximum matching. If no $Y \in \mathbf{Y}$ is connected with nodes in $\mathbf{N}_{\mathbf{V}_{\overline{m}}}$, there must be some nodes in $\mathbf{Y}$ connected with nodes $\mathbf{X}' \subseteq \mathbf{N}_{\mathbf{V}_m}$ and $\mathbf{X}' \nsubseteq \mathbf{X}$. We identify the nodes $\mathbf{Y}'$ matched with $\mathbf{X}'$ in $\mathbb{A}_m$. Let $\mathbf{Y} = \mathbf{Y} \cup \mathbf{Y}'$ and $\mathbf{X} = \mathbf{X} \cup \mathbf{X}'$. Apply the same argument that $\mathbf{Y}$ must connect with nodes other than $\mathbf{X}$ in $\mathbb{A}$; otherwise, $|\mathbf{Y} \cup N| > |\mathbf{X}|$, contradicting with the bipartite graph representation of a structure. We can then repeat our previous reasoning and follow the same argument. Since we have finite number of nodes in $BG(\mathbf{E})$, We will eventually have a node in $Y \in \mathbf{Y}$ that connects to a node $Z \in \mathbf{N}_{\mathbf{V}_{\overline{m}}}$. We can then replace the matched arcs backward along the path of extension and finally reach N to increase the cardinality of $\mathbb{A}_m$. We prove by contradiction that there exist no maximum matching $\mathbb{A}_m$ of $BG(\mathbf{E})$ with $\mathbf{N}_{\mathbf{E}_{\overline{m}}} \neq \emptyset$. □

Although all $\mathbf{N}_{\mathbf{E}} = \mathbf{N}_{\mathbf{E}_m}$ are matched to $\mathbf{N}_{\mathbf{V}_m}$ in $\mathbb{A}_m$ according to Lemma 3.2, an arc (N_{e_i}, N_{V_j}) in $\mathbb{A}_m$, where $N_{e_i} \in \mathbf{N}_{\mathbf{E}_m}$ and $N_{V_j} \in \mathbf{N}_{\mathbf{V}_m}$, does not necessarily represent a direct dependency relations as an arc in a complete matching for the $BG(\mathbf{E})$ of a self-contained structure $\mathbf{E}$. The reason is that the values of unmatched variables $\mathbf{N}_{\mathbf{V}_{\overline{m}}}$ are not determined by any equations in $\mathbb{A}_m$. Therefore, they cannot be used by $\mathbf{X} \in \mathbf{N}_{\mathbf{E}_m}$, the set of matched equations connected with $\mathbf{N}_{\mathbf{V}_{\overline{m}}}$ in $\mathbb{A}$, to determine the values of variables represented by the corresponding matched nodes $\mathbf{Y}$. In other words, those matchings composed by $\mathbf{X}$ and $\mathbf{Y}$ do not represent valid direct dependency relations. I can then view the values of $\mathbf{Y}$ as undermined and apply the same reasoning again and again to extend $\mathbf{X}$ and $\mathbf{Y}$ until all invalid direct dependency relations are identified. More precisely, I propose the following procedure that modifies a maximum matching $\mathbb{A}_m$ of the $BG(\mathbf{E})$ of an under-constrained structure $\mathbf{E}$ into a matching $\mathbb{A}_{m'}$ in which each arc represents a direct dependency relation.

1. $\mathbb{A}_{m'} := \mathbb{A}_m$; $\mathbf{Y} := \mathbf{N}_{\mathbf{V}_{\overline{m'}}}$.
2. Identify the set of nodes $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{E}_{m'}}$ that are adjacent to $\mathbf{Y}$ in $\mathbb{A}$.
3. If $\mathbf{X} = \emptyset$, return $\mathbb{A}_{m'}$.
4. Identify the set of nodes $\mathbf{Z} \subseteq \mathbf{N}_{\mathbf{V}_{m'}}$ that are matched to $\mathbf{X}$ in $\mathbb{A}_{m'}$. Let $\mathbb{A}_r \subseteq \mathbb{A}_{m'}$ denote the set of arcs that match $\mathbf{Z}$ and $\mathbf{X}$.

5. Let $\mathbb{A}_{m'} := \mathbb{A}_{m'} \setminus \mathbb{A}_r$. Consequently, $\mathbf{Z} \subseteq \mathbf{N}_{\mathbf{V}_{\overline{m'}}}$ and $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{E}_{\overline{m'}}}$.

6. Let $\mathbf{Y} := \mathbf{Z}$ and go to Step 2.

Given a finite under-constrained structure $\mathbf{E}$, the procedure will stop and return $\mathbb{A}_{m'}$. If $\mathbb{A}_{m'} = \emptyset$, then $\mathbf{E}$ is strictly under-constrained because there is no direct dependency relation in $\mathbf{E}$, i.e., $\mathbf{E}$ does not contain any self-contained structure. If $\mathbb{A}_{m'} \neq \emptyset$, then $\mathbf{N}_{\mathbf{E}_{m'}}$ represents a self-contained structure $\mathbf{E}_{m'}$ and $\mathbf{N}_{\mathbf{E}_{\overline{m'}}}$ represents the derived strictly under-constrained structure $\mathbf{E}_{\overline{m'}}$.

Since there exist more than one maximum matching for the $BG(\mathbf{E})$ of an under-constrained structure $\mathbf{E}$, I prove Theorem 3.4 to show that the proposed procedure derives the same $\mathbf{N}_{\mathbf{E}_{m'}}$ and $\mathbf{N}_{\mathbf{E}_{\overline{m'}}}$ given any maximum matching $\mathbb{A}_m$. Consequently, the theorem derives unique $\mathbf{E}_{m'}$, the union of self-contained structures embedded in $\mathbf{E}$, and $\mathbf{E}_{\overline{m'}}$, derived strictly under-constrained structure, from any maximum matching $\mathbb{A}_m$.

Theorem 3.4

Any maximum matching $\mathbb{A}_m$ of the bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N}_\mathbf{E}, \mathbf{N}_\mathbf{V}, \mathbb{A} \rangle$ of an under-constrained structure $\mathbf{E}$ has the same $\mathbf{N}_{\mathbf{E}_{m'}}$ and $\mathbf{N}_{\mathbf{E}_{\overline{m'}}}$ with respect to the modified matching $\mathbb{A}_{m'}$.

Proof: By contradiction. Assume that there are two different maximum matchings $\mathbb{A}_{m_1}$ and $\mathbb{A}_{m_2}$, for which the procedure returns two different modified matchings $\mathbb{A}_{m'_1}$ and $\mathbb{A}_{m'_2}$ with different $\mathbf{N}_{\mathbf{E}_{m'_1}}$ and $\mathbf{N}_{\mathbf{E}_{m'_2}}$ (similarly $\mathbf{N}_{\mathbf{E}_{\overline{m'_1}}}$ and $\mathbf{N}_{\mathbf{E}_{\overline{m'_2}}}$). Since both modified matchings contain all direct dependency relations encoded in $\mathbf{E}$ according to the procedure, we know that $\mathbb{A}_{m'_1}$ and $\mathbb{A}_{m'_2}$ are complete matchings for subgraphs $BG(\mathbf{E}_{m'_1})$ and $BG(\mathbf{E}_{m'_2})$ of $BG(\mathbf{E})$ respectively. According to Theorem 3.3, both $\mathbf{E}_{m'_1} \subset \mathbf{E}$ and $\mathbf{E}_{m'_2} \subset \mathbf{E}$ are self-contained structures in $\mathbf{E}$. However, Theorem 3.2 implies that there is only one derived strictly under-constrained structure in an under-constrained structure $\mathbf{E}$. Therefore, $\mathbf{E} \setminus \mathbf{E}_{m'_1} = \mathbf{E} \setminus \mathbf{E}_{m'_2}$ and $\mathbf{E}_{m'_1} = \mathbf{E}_{m'_2}$. It contradicts $\mathbf{E}_{m'_1} \neq \mathbf{E}_{m'_2}$ that derives from our assumption $\mathbf{N}_{\mathbf{E}_{m'_1}} \neq \mathbf{N}_{\mathbf{E}_{m'_2}}$. □

I remark that not all modified matchings $\mathbb{A}_{m'}$ are the same given different maximum matchings $\mathbb{A}_m$ of an under-constrained structure $\mathbf{E}$. As long as there are no strongly-coupled components in $\mathbf{E}_{m'}$ with respect to $\mathbb{A}_{m'}$, any maximum matching $\mathbb{A}_m$ results in the same modified matching $\mathbb{A}_{m'}$.

I formalize the extended causal ordering algorithm that based on maximum matching in bipartite graph, denoted as $ECOA_{BGM}$, in Figure 3.6 and Figure 3.7. $ECOA_{BGM}$ takes a structure $\mathbf{E}$

Procedure $ECOA_{BGM}(\mathbf{E})$

Input: A structure matrix $\mathbf{E}$.

Output: A graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$, where $\mathbf{N}$ represents $\mathbf{Vars}(\mathbf{E})$, $\mathbf{A}$ is a set of directed arcs, and $\mathbb{A}$ is a set of undirected arcs among $\mathbf{N}$.

1. Create the bipartite graph $BG(\mathbf{E}) = \langle \mathbf{N_E}, \mathbf{N_V}, \mathbb{A}_{BG} \rangle$.
2. Identify a maximum matching $\mathbb{A}_m \subset \mathbb{A}_{BG}$ of $BG(\mathbf{E})$.
3. **if** $|\mathbb{A}_m| < |\mathbf{N_V}|$
4. $\quad \mathbf{Y} := \mathbf{N}_{\mathbf{V}_{\overline{m}}}$; Identify $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{E}_m}$ that are adjacent to $\mathbf{Y}$ in $\mathbb{A}_{BG}$.
5. $\quad$ **do**
6. $\quad\quad$ Identify $\mathbf{Z} \subseteq \mathbf{N}_{\mathbf{V}_m}$ that are matched to $\mathbf{X}$ in $\mathbb{A}_m$.
7. $\quad\quad$ Let $\mathbb{A}_r \subseteq \mathbb{A}_m$ denote the set of arcs that match $\mathbf{Z}$ and $\mathbf{X}$.
8. $\quad\quad$ $\mathbb{A}_m := \mathbb{A}_m \setminus \mathbb{A}_r$. Consequently, $\mathbf{Z} \subseteq \mathbf{N}_{\mathbf{V}_{\overline{m}}}$ and $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{E}_{\overline{m}}}$.
9. $\quad\quad$ $\mathbf{Y} := \mathbf{Z}$; Identify $\mathbf{X} \subseteq \mathbf{N}_{\mathbf{E}_m}$ that are adjacent to $\mathbf{Y}$ in $\mathbb{A}$.
10. $\quad$ **until** $\mathbf{X} = \emptyset$.
11. **end if**
12. Construct a graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A}, \mathbb{A} \rangle$ from $\mathbb{A}_m$, where
13. $\quad$ the set of nodes $\mathbf{N}$ corresponding to $\mathbf{N_V}$, and
14. $\quad$ **for each** arc (N_{e_i}, N_{V_j}) in $\mathbb{A}_m$
15. $\quad\quad$ construct a set of directed arcs $N_k \rightarrow N_j$
16. $\quad\quad\quad$ for each variable $V_k \in \{\mathbf{Vars}(e_i) \setminus \{V_j\}\}$ to variable V_j.
17. $\quad$ **end for each**

Continued on Figure 3.7.

Figure 3.6. Extended causal ordering algorithm based on bipartite graph matching.

as input and outputs a causal graph $G(\mathbf{E})$ for a self-contained structure or a graph $G(\mathbf{E})$ depicting causal and relevant relations for an under-constrained structure. $ECOA_{BGM}$ first construct the bipartite graph $BG(\mathbf{E})$ and then identify a maximum matching $\mathbb{A}_m$ of $BG(\mathbf{E})$. $ECOA_{BGM}$ checks if $\mathbb{A}_m$ is a complete matching. If not, $ECOA_{BGM}$ modifies $\mathbb{A}_m$, as listed in Line 3–11, such that each arc in $\mathbb{A}_m$ represents a direct dependency relation. $ECOA_{BGM}$ then constructs a graph with

Continued from Figure 3.6.

18. **if** ($\mathbf{N}_{\mathbf{E}_{\overline{m}}} \neq \emptyset$)
19. **for each** $N_e \in \mathbf{N}_{\mathbf{E}_{\overline{m}}}$
20. Let $\mathbf{N}_{\mathbf{X}} \subseteq \mathbf{N}_{\mathbf{V}_m}$ and $\mathbf{N}_{\mathbf{Y}} \subseteq \mathbf{N}_{\mathbf{V}_{\overline{m}}}$ connected with N_e in $\mathbb{A}_{BG}$.
21. Construct a set of directed arcs $N_i \rightarrow N_j$
22. for each pair of $N_{X_i} \in \mathbf{N}_{\mathbf{X}}$ and $N_{Y_j} \in \mathbf{N}_{\mathbf{Y}}$.
23. Construct a set of pair-wised undirected arcs among
24. nodes corresponding to $\mathbf{N}_{\mathbf{Y}}$ in $\mathbf{N}$.
25. **end for each**
26. **end if**
27. Identify the strongly-connected components $\mathbf{S}$ in $G(\mathbf{E})$.
28. **for each** strongly-connected component $S_i \in \mathbf{S}$
29. **if** $|S_i| > 1$
30. Collect the set of parents of each $N_j \in S_i$ as $\mathbf{P}$.
31. Remove all incoming arcs of each $N_j \in S_i$.
32. Create a partition $\mathbf{N}_i$ with all nodes in S_i.
33. Draw $N_j \in \mathbf{N}_i$ as a cloud for the strongly-coupled component.
34. Create directed arcs $P_i \rightarrow \mathbf{N}_i$ for all $P_i \in \mathbf{P}$.
35. **end if**
36. **end for each**
37. Return $G(\mathbf{E})$.

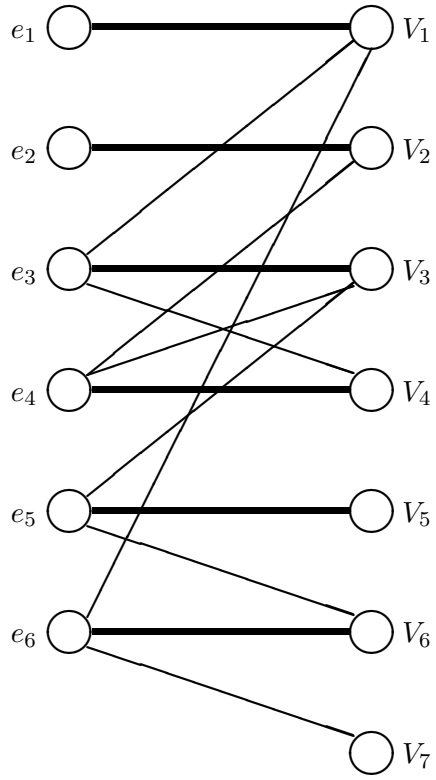
Figure 3.7. Extended causal ordering algorithm based on bipartite graph matching (continued).

directed arcs with respect to $\mathbb{A}_m$. In Line 12–26, $ECOA_{BGM}$ depicts causal relations and relevant relation among the derived strictly under-constrained structure. $ECOA_{BGM}$ then identifies the strongly connected component and modifies the graph for the strong coupled components in Line 28–36. $ECOA_{BGM}$ is a worst-case polynomial time algorithm, since all of the codes in Figure 3.6 and Figure 3.7 can be done in linear time except for finding the maximum matching in Line 2 and modifying maximum matching in Line 3–11 require polynomial time.

Example 3.2 Consider applying the $ECOA_{BGM}$ algorithm to the structure in Figure 3.8 (a). $ECOA_{BGM}$ first construct the corresponding bipartite graph of the structure as shown in Figure 3.8 (b). Next, $ECOA_{BGM}$ computes the maximum matching as shown in Figure 3.8 (b), where each bold arc depicts an arc in the matching. $ECOA_{BGM}$ then modifies the maximum matching as shown in Figure 3.9 (a). $ECOA_{BGM}$ finally creates the graph, as shown in Figure 3.9 (b), that depicts causal and relevant relations in the under-constrained structure in Figure 3.8 (a). □

	V_1	V_2	V_3	V_4	V_5	V_6	V_7
e_1	x	0	0	0	0	0	0
e_2	0	x	0	0	0	0	0
e_3	x	0	x	x	0	0	0
e_4	0	x	x	x	0	0	0
e_5	0	0	x	0	x	x	0
e_6	0	x	0	0	0	x	x

(a)



(b)

Figure 3.8. (a) The bipartite graph representation of an under-constrained structure **E**. (b) A maximum matching of the bipartite graph is depicted as bold arcs.

3.6 Interactive Modeling Process

The modeling process starts with an initial focus, which is normally, in the spirit of value-focused thinking [Keeney, 1994], the value variable. Users can also start with other focus variables such as decision, observation, or whatever criteria that is pertinent or important a-priori. Using *ImaGeNIe*

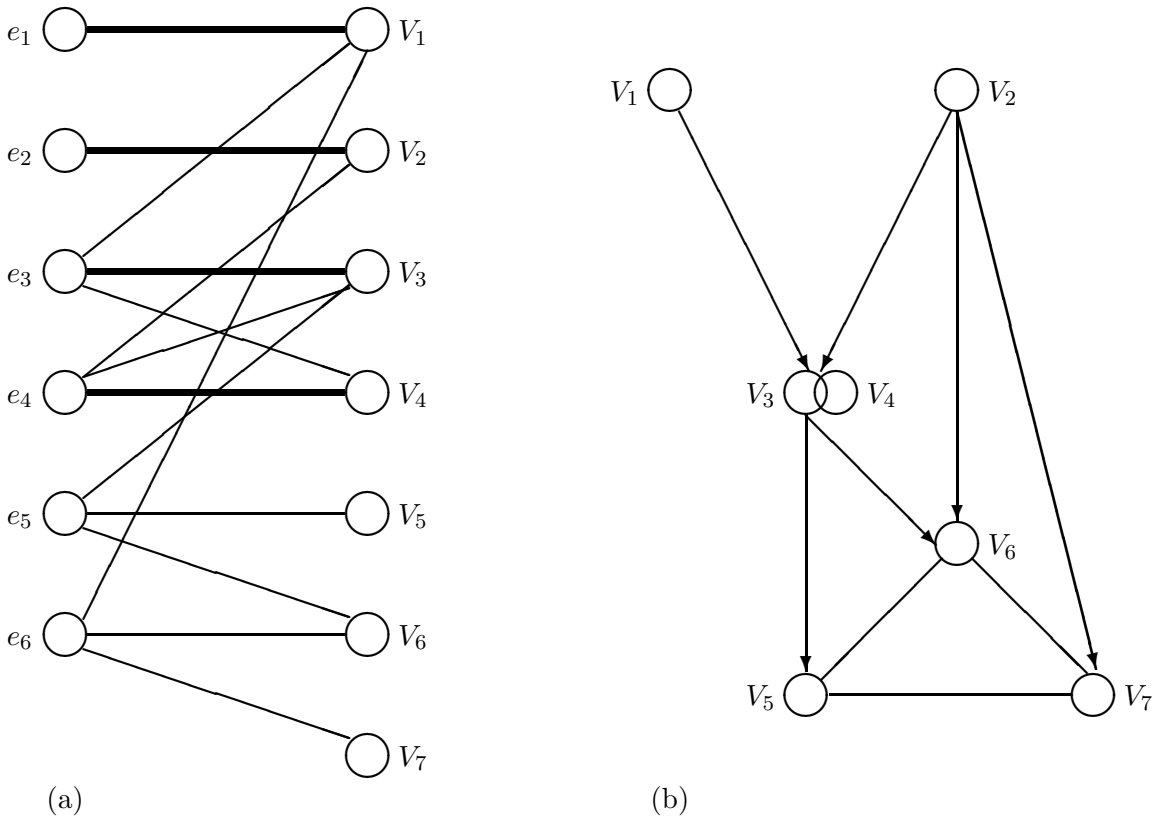


Figure 3.9. (a) The modified matching of Figure 3.8 (b). $ECOA_{BGM}$ outputs the graph in (b) to depict causal and relevant relations among variables in the under-constrained structure in Figure 3.8 (a).

environment, users can interactively browse the mechanisms related to their focus variables in mechanism knowledge base. A key word search interface for mechanisms is provided in mechanism knowledge base to facilitate finding relevant mechanisms given the name of a variable. Users select mechanisms that best depict the problem at hand, bring them into workspace, merge them, or specify exogenous variables to set the boundary of the system. However, I suggest the users to focus on one variable and add relevant mechanisms one at a time as the model evolves, since it resembles the action of focusing on a variable of interest, explaining or observing it in terms of its underlying mechanism. The users repeat the process iteratively until the model is requisite. In other words, users make decisions on the level of granularity and when to stop in the model building process. The system only plays the passive role of an assistant: assisting in searches for relevant mechanisms, indicating the possible mechanisms to merge, denoting the manipulable variables, and showing the status of each variable and (causal or relevant) relations among variables in the

workspace.

A model is normally evolved from an under-constrained model to a self-contained model. Designating manipulable variables as exogenous helps in obtaining a self-contained system, i.e., orienting all arcs in the graph. If the user assigns a potential policy variable, a manipulable variable that is endogenous in a self-contained system, as exogenous, the whole model becomes over-constrained, because the number of equations is greater than the number of variables. I allow a model to be under-constrained or self-contained at any stage of the model development in *ImaGeNIe*, but I disallow a model to be over-constrained. When a model becomes over-constrained, the system present a list of mechanisms that are currently in the model and asks users to release one of them in order to change the system into a self-contained or an under-constrained system. The implication of such manipulations is further discussed in Chapter 4.

3.7 Example Model Building Session

The University Performance Budget Planning Model [Simon *et al.*, 2000] is composed of thirty-eight nonlinear structural equations that describe interactions among eighty-eight variables. These structural equations are divided into seven subsystems: *Teaching Operations*, *Teaching Expenditures*, *Research Expenditures*, *Income*, *Space Cost*, *Total Expense*, and *Surplus*. I encode these structural equations into a mechanism knowledge base and demonstrate how to use *ImaGeNIe* to build a simplified university budget model.

Suppose Tom, an officer of the budget planning office, would like to plan the expense on the faculty salary for next year. He starts up *ImaGeNIe* and loads the university knowledge base. He may use the *navigation tree* to locate the relevant mechanisms for *faculty salary*. Suppose he identifies a mechanism that describes the interactions among variables: faculty salary (*facsal*), other income (*oinc*), tuition fee (*tuition*), number of students (*nstud*), number of faculty (*nfac*), and overhead (*overh*), as $facsal = (oinc + tuition * nstud)/(nfac * (1 + overh))$. He selects the mechanism by clicking and dragging it into the model building workspace. The extended causal ordering module generates the corresponding graph as shown in Figure 3.10 and the variables in the workspace are also shown in the network tree view. He then designates *nstud*, *nfac*, *tuition*, *overh*, and *oinc* as exogenous variables by either right-clicking on the context menu of the nodes

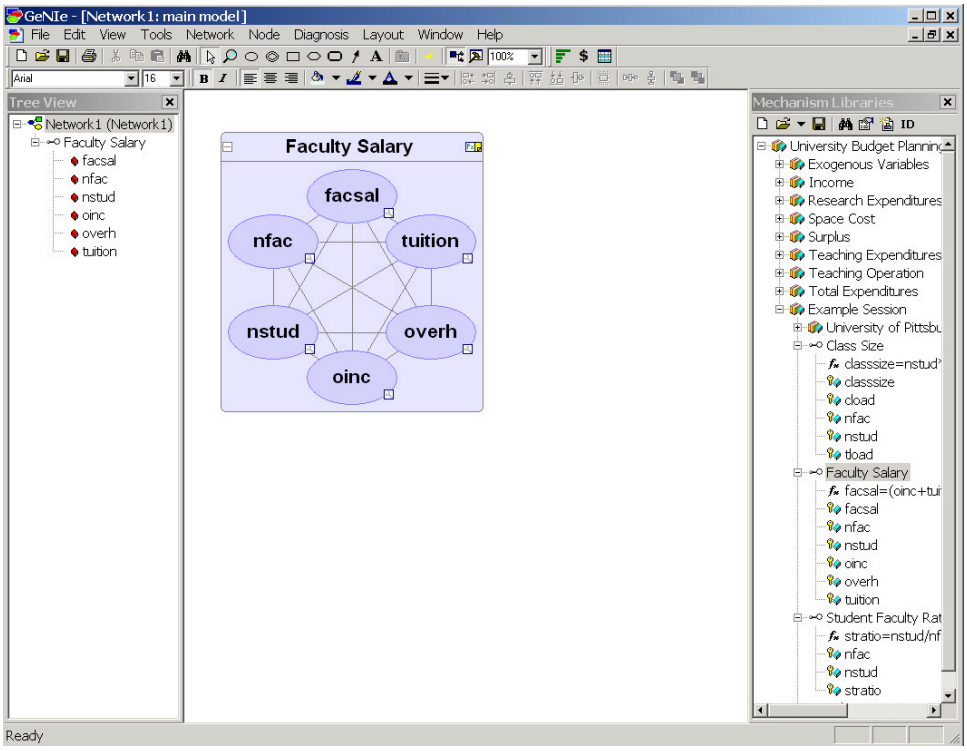


Figure 3.10. Model builder selects the mechanism relevant to the faculty salary into workspace and the causal ordering generates the corresponding graph.

in the workspace or the nodes in network tree view to invoke the *Make exogenous* dialogue box to provide values for those exogenous variables. The extended causal ordering module then derives the new graph (See Figure 3.11). He then identifies that the student-faculty ratio (*stratio*) is also an relevant mechanism which is described as $stratio = nstud/nfac$ and brings it into the workspace. The workspace of *ImaGeNIe* now contains two mechanism boxes, representing two mechanisms brought in by Tom, where each mechanism box has a title named by the mechanism: faculty salary and student-faculty ratio. He can then integrate these two mechanisms by selecting the node *Number of students* (*nstud*) from one (source) mechanism box and drag it over to the other (target) mechanism box. When the mouse cursor enters the target mechanism box, *ImaGeNIe* search over identifiers in the target mechanism box such that the variable with the identifier same as the source node is automatically highlighted. He can then release the mouse cursor over the highlighted variable to merge the two variables. *ImaGeNIe* then automatically merges other variables with the

same identifiers in both mechanism boxes and integrates two mechanism boxes into one box with two mechanisms (See Figure 3.12). He then brings in the mechanism describing the interactions among variables: *class size* (*classize*), *number of students* (*nstud*), *class load* (*cload*), *number of faculty* (*nfac*), and *teaching load* (*tload*), as $classsize = (nstud * cload)/(nfac * tload)$. He merges the mechanism boxes of class size with the integrated mechanism box. He then makes *teaching load* (*tload*) and *class load* (*cload*) exogenous and obtains a self-contained model that describes the causal relations among those variables of interests (See Figure 3.13). He can now read off the following causal relations from the self-contained model:

- *Faculty salary* is determined by the *number of students*, the *number of faculty*, *tuition fee*, *other income*, and *overhead*.
- *Student-teacher ratio* is determined by the *number of students* and the *number of faculty*.
- *Class size* is determined by the *number of students*, the *number of faculty*, *class load*, and *teaching load*.

3.8 Discussion

Support for building model structure is one of the best ways to improve the quality of advice based on decision-theoretic models. I believe that human judgement with respect to relevance, model size, completeness, and granularity is more reliable. While existing approaches focus on automatic model construction either from knowledge base or directly from data, the approach proposed in this dissertation favors a closely-coupled loop between the system and its user. Built on the assumption that under-constrained models reflect the problem recognition stages, *ImaGeNIe* assists users in encoding their conceptual problem in a causal graph generated by the extended causal ordering algorithm. Furthermore, *ImaGeNIe* provides users with the flexibility to choose building blocks from a knowledge base to extend the model, to manipulate the variables, and to extract reusable mechanisms from existing models to knowledge bases. The concept of causal mechanisms, on which *ImaGeNIe* relies, provides a general means to accommodate different forms of knowledge description and makes the knowledge acquisition task easier. In addition, *ImaGeNIe*

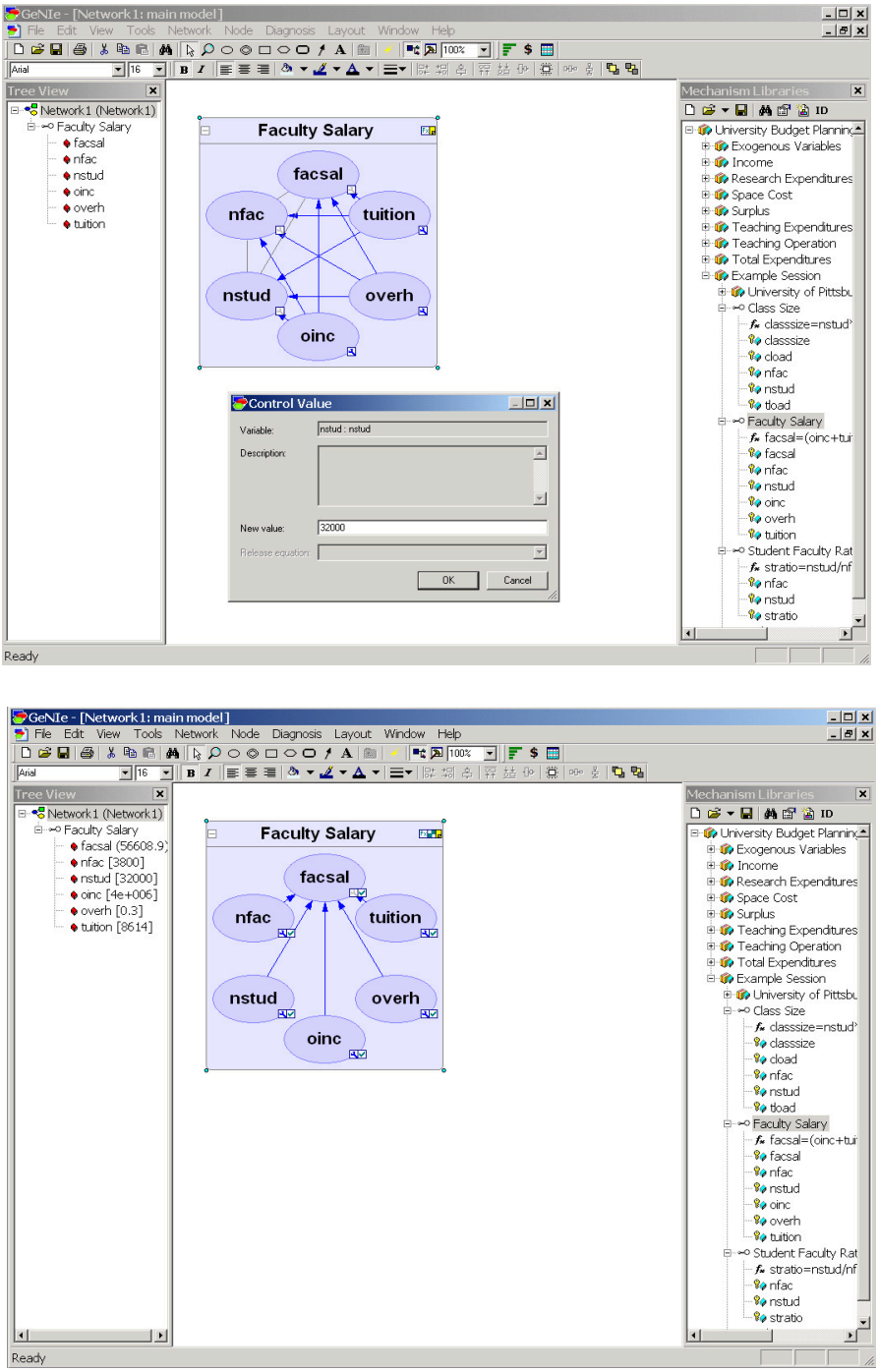


Figure 3.11. Model builder makes variables *number of students* (*nstud*), *number of faculties* (*nfac*), *tuition* (*tuition*), *overhead* (*overh*), and *other income* (*oinc*) become exogenous (top) and the causal ordering generates the corresponding graph (bottom).

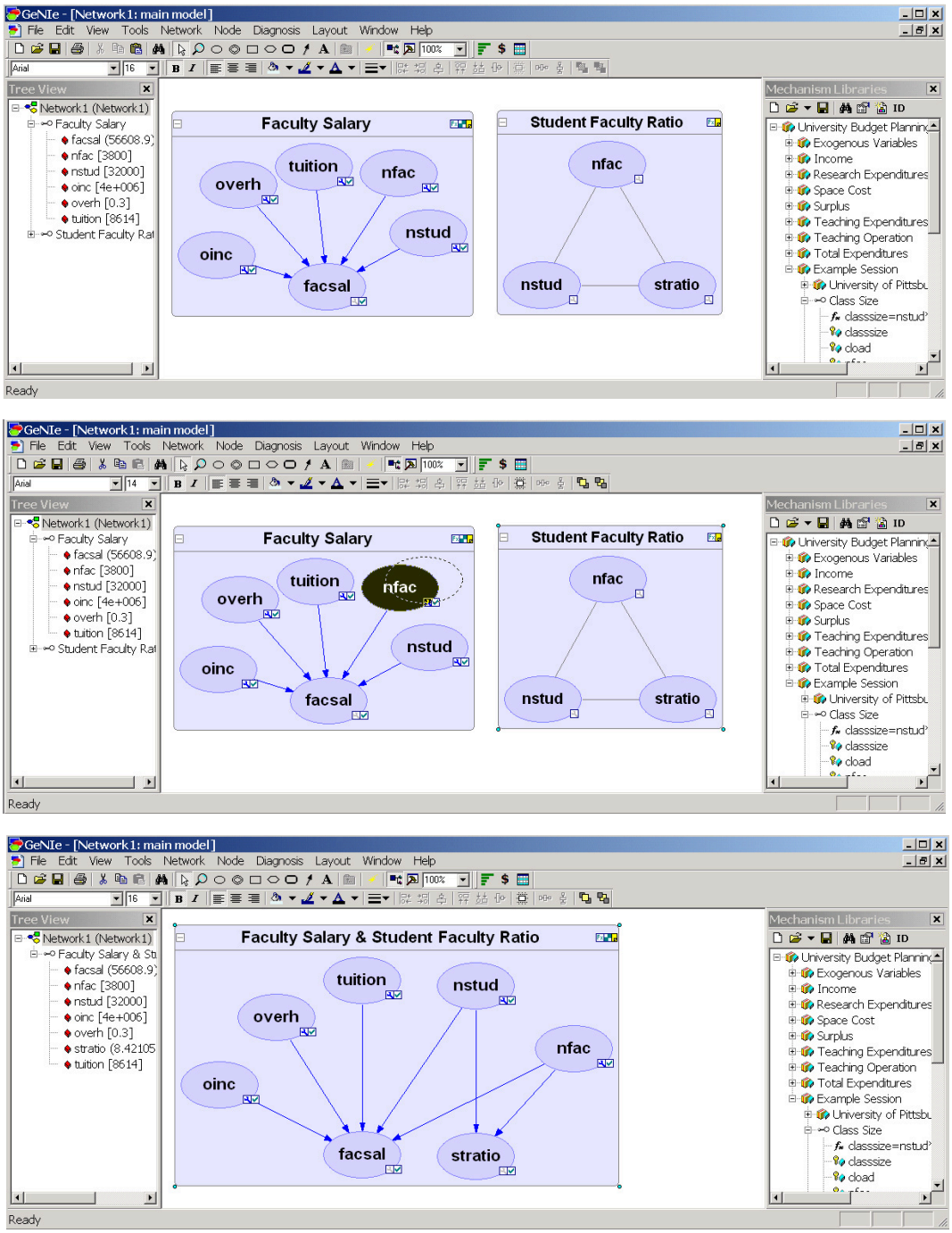


Figure 3.12. Model builder drops the *student-teacher ratio* mechanism into workspace (top) and perform the merge operations for *number of students* (*nstud*) and the *number of faculty* (*nsfac*) (middle). The causal ordering algorithm generates the corresponding graph (bottom).

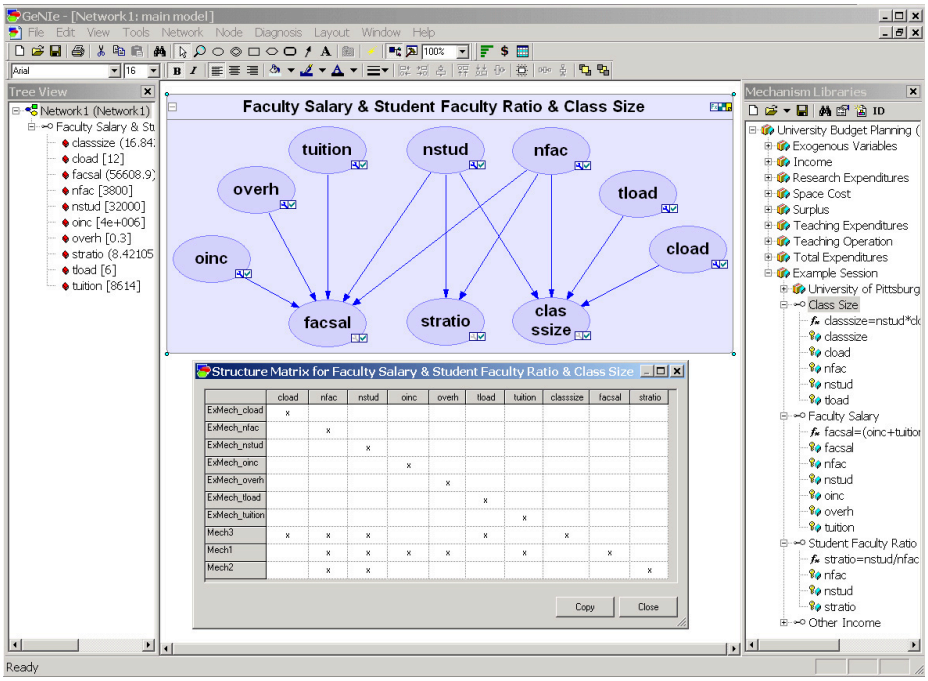


Figure 3.13. A simplified university model that contains three core mechanisms and seven value assignment equations. The causal graph of the model is shown on the top; and the structure matrix of the model is shown at the bottom.

helps users to organize causal mechanisms into subsystems so that users can effectively access the knowledge base.

Recent research in applying the object-oriented framework to extend Bayesian networks for modeling complex domains [Koller and Pfeffer, 1997; Laskey and Mahoney, 1997; Pfeffer *et al.*, 1999] are closely related to this work. Each of these approaches organizes domain knowledge into a hierarchical system. In Object-Oriented Bayesian Networks (OOBN), the domain knowledge is structured explicitly as a class-hierarchy for the type system and as an object-hierarchy for the real model. In my framework, I do not impose any constraints on how users should organize their domain knowledge in the knowledge base. In the future, I would like to explore the semantics for combining the type system with causal mechanisms so that the knowledge base can more efficiently store the domain knowledge and be more effectively used by users. As for the constructed models, *ImaGeNIe* provides submodels to group nodes into a graphical organization unit for the sake of succinct presentation, but there is no special semantic meaning attached to submodels in terms

of inference. In the future, I plan to impose d-sepset [Xiang *et al.*, 1993] constraints on submodel composition such that each submodel has well defined I/O sets to resemble object hierarchy in OOBN.

Once the model structures generated from the framework are associated with variable ranges and their numerical parameters, such as explicit equations or conditional probability tables (CPTs), manipulation on the model may invalidate these numerical parameters. Druzdzel and van Leijen [2001] have shown the special conditions under which the CPTs in Bayesian networks can be reversed under manipulation. As for the explicit equations, *ImaGeNIe* tries to solve the manipulated system symbolically if there exists a solution. I would like to further explore conditions under which one can derive the numerical parameters for the manipulated models.

ImaGeNIe provides a flexible interactive model building environment for users to build models in causal form with as much system assistance as possible but without giving up their control over the model building process. I believe my efforts in incorporating causality as a heuristic in aiding model building and knowledge acquisition is an important extension to the existing approaches. The experimental results on the effectiveness of *ImaGeNIe* will be presented in Chapter 6.

Chapter 4
Changes in Structure

The term "changes in structure," originating from work in econometrics, refers to structural modifications resulting from modeling the effects of actions on a causal model. In this Chapter, I address the problem of changes in structure in systems containing reversible mechanisms. A mechanism is reversible if the causal relations among its variables change as the mechanism is embedded in different systems, or in different operational contexts of a system. I formalize the representation of the reversibility property of a mechanism for supporting the modeling of changes in structure in systems containing reversible mechanisms. Causal models built on my formalization can answer two new types of queries: (1) When manipulating a potential policy variable (unmanipulated manipulable variable) in a causal model (i.e., making an endogenous variable exogenous), which structural equations are possibly invalidated and can be removed from the model? (2) Which potential policy variables may be manipulated in order to invalidate and, effectively, remove a structural equation from a model?

Section 4.1 introduces the problem of changes in structure. Section 4.2 introduces my formalization of the reversibility of a mechanism. Section 4.3 discusses different representations of actions and introduces action operator $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ defined upon the level of mechanisms. Section 4.4 introduces the problem of action deliberation and provides algorithms to assist in forming atomic actions. Section 4.5 presents the support of changes in structure in *ImaGeNIe*.

4.1 Introduction

The problem of predicting the effects of actions was originally studied in econometrics literature as the problem of *changes in structure* in simultaneous structural equation models. Assuming that a modeler has sufficient prior knowledge to predict the effects of actions, researchers in econometrics modeled the effects of actions by "scraping" invalid equations and "replacing" them by new ones [Marschak, 1953; Simon, 1953; Wold, 1954; Strotz and Wold, 1960]. If we assume that the variable

manipulated by an action is governed by an *irreversible* mechanism, the effect of an action amounts to the arc-cutting operation in the causal graph describing the system [Pearl, 2000; Spirtes *et al.*, 1993]. For example, rain (R) can make us wet (W); wearing a rain coat will protect us from getting wet but it does not make rain go away. To model this, we cut the arc $R \rightarrow W$, since the mechanism between rain (R) and getting wet (W) is irreversible. However, there exists a large class of *reversible* mechanisms [Simon, 1953; Wold and Jureen, 1953; Wold, 1954; Simon and Rescher, 1966; Druzdzel, 1992; Spirtes *et al.*, 1993; Pearl, 2000] that are not amenable to the arc-cutting operation. For example, a car engine causes wheels to turn when going up hill, but the wheels slow down the engine when going down hill with transmission being put in a lower gear. An action may reverse the direction of causal relations among variables and consequently have drastic effects on causal graphs.

There have been attempts to assist in predicting the effects of actions on systems containing reversible mechanisms. Bogers [1997] developed theorems to support structure modifications in which the equation being removed by an action governs an exogenous variable. Druzdzel and van Leijen [2001] studied the conditions under which a conditional probability table in a causal Bayesian network can be reversed when manipulating a reversible mechanism. Dash and Druzdzel [2001] demonstrated how various recursive equilibrium systems may violate the *Manipulation Postulate*, arc-cutting operation, when actions are applied on equilibrium systems; they attributed such violations to the problem of reversible mechanisms.

My approach to supporting changes in structure is based on an explicit representation of the reversibility of a mechanism. I define the reversibility of a mechanism semantically on the set of possible effect variables of a mechanism. A set of structural equations is a causal model only if the causal relations among the variables are consistent with the reversibility of its mechanisms. Similarly to STRIPS language [Fikes and Nilsson, 1971], I represent an action as $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$, where $\mathbf{E}$ is the model that an action applies on, $\mathbf{E_{pre}}$ is the set of preconditions that needs to be satisfied before an action can be applied, $\mathbf{E_{add}}$ is the set of structural equations to be added into $\mathbf{E}$, and $\mathbf{E_{del}}$ is the set of structural equations to be removed from $\mathbf{E}$. Once an action is completely specified, the effect of an action is simply performing the modifications specified in $\mathbf{E_{add}}$ and $\mathbf{E_{del}}$ lists on the causal model $\mathbf{E}$ when $\mathbf{E_{pre}}$ is satisfied. Given the $\mathbf{E_{pre}}$ and one of the $\mathbf{E_{add}}$ or $\mathbf{E_{del}}$ lists of a partially specified action, I prove two theorems to assist modelers in answering two new types

of queries: (1) When manipulating a potential policy variable in a causal model, which structural equations are possibly invalidated and can be removed from the model? (2) Which potential policy variables may be manipulated in order to invalidate and, effectively, remove a structural equation from a model? My approach enables us to predict the effects of actions on systems consisting of mixtures of mechanisms.

4.2 Reversible Mechanisms and Reversibility

Traditionally, reversible mechanisms are discussed mainly in deterministic mechanical and physical relations [Wold and Jureen, 1953, pp. 325], since the invertibility of a function is a necessary condition for the reversibility. A functional relation may be reversible in *functional* sense, but may not be reversible in *causal* sense [Wold, 1954, footnote 6]. For example, ideal gas law and Ohm's law are given in [Wold and Jureen, 1953, pp. 40] and [Nayak, 1994, pp. 10] respectively as examples of only partially reversible mechanisms, even though their functional relations are invertible. Wold [1964, pp. 279] claimed that a deterministic relation may or may not be causally reversible, but a stochastic relation never is. However, Druzdzel and van Leijen [2001] demonstrated that under some special conditions probability distribution tables in a causal Bayesian network can be reversed in both functional and causal senses.

I propose to explicitly represent the reversibility of a mechanism to assist model construction and predictions of the effect of actions on systems containing reversible mechanisms. I define the reversibility of a mechanism semantically on the set of possible effect variables of a mechanism. In my framework, modelers are allowed to associate more than one causal relation to a mechanism with reasonable a-priori assumptions. Modelers start with identifying variables in a mechanism qualitatively, then expressing a causal relation qualitatively as a specification of the effect variable and its causes, and, finally, giving such qualitative specification an explicit quantitative specification in a functional form. Assuming that the number of variables in a mechanism is finite, the number of possible effect variables for a mechanism is also finite. I can classify mechanisms into four categories according to their *reversibility*: (1) *completely reversible* (CR): every variable in the mechanism can be an effect variable, (2) *partially reversible* (PR): some of the variables in the mechanism can be effect variables, (3) *irreversible* (IR): exactly one of the variables in the mechanism can be an

effect variable, and (4) *unknown* (*UN*): the reversibility of the mechanism is unspecified, i.e., the modeler only asserts that variables in a mechanism are relevant, but has not yet resolved how they relate to each other causally.

Definition 4.1 (reversibility)

Let $\mathbf{Vars}(e)$ *be the set of variables in a structural equation* e*. Let* $\mathbf{EfVars}(e) \subseteq \mathbf{Vars}(e)$ *be the set of all possible effect variables in a structural equation* e*. The reversibility of a mechanism represented by* e *is*

1. completely reversible *if* $\mathbf{EfVars}(e) = \mathbf{Vars}(e)$ *and* $|\mathbf{EfVars}(e)| > 1$,
2. partially reversible *if* $1 < |\mathbf{EfVars}(e)| < |\mathbf{Vars}(e)|$,
3. irreversible *if* $|\mathbf{EfVars}(e)| = 1$, *and*
4. unknown *if* $|\mathbf{EfVars}(e)| = \emptyset$.

I emphasize that the notion of reversibility of a mechanism is a semantic one since it is defined with respect to the set of effect variables of a mechanism with a-priori assumptions. In addition, reversibility is defined as the property of a mechanism, but not as a derived property of a mechanism when it is embedded in a system. In other words, the set of effect variables of a mechanism is assumed a-priori before we decide which system the mechanism will be embedded in. Which effect variable is *active* will be determined as soon as we know which system the mechanism is embedded in.

Notice that the notion of entity plays an essential role in our modeling. The notion of reversibility of a mechanism is very often confused with *causal mixtures* [Cooper, 1999] in which members of entities may not share the same causal relationships. For example, if the relation between schooling and verbal ability is modeled as a causal mixture, we may find that schooling helps to increase verbal ability in one subpopulation of students but verbal ability helps to get higher schooling in another. However, reversible mechanisms model the same entities in different contexts. For example, the verbal ability helps some population of students to get higher schooling in one context, but in another context the schooling helps the same students to increase their verbal ability.

To assist modeling with systems containing reversible mechanisms, I explicitly represent the set of possible effect variables of each structural equation, along with their explicit functional forms, in

a mechanism knowledge base as presented in Chapter 3. In Section 2.4, I assumed that all structural equations in a structural equation model described causal mechanisms active in the system. When predicting the effects of actions on systems containing reversible mechanisms, I need to verify if a manipulated model, derived for a queried intervention, is still consistent with the prior knowledge stored in the knowledge base. I define *consistency* as follows.

Definition 4.2 (consistency)
A set of structural equations $\mathbf{E} = \{e_1, e_2, \ldots, e_m\}$ *is* consistent *with a knowledge base* $\mathcal{K}$ *if there exists* $\mathbf{F} = \{f_1, f_2, \ldots, f_m\}$*, where each* $f_i \in \mathbf{F}$ *is an explicit functional form of* $e_i \in \mathbf{E}$*, stored in* $\mathcal{K}$ *such that* $\mathbf{F}$ *is a self-contained structure.*

Given a set of structural equations $\mathbf{E}$, I can test if $\mathbf{E}$ is consistent with $\mathcal{K}$ as described in the procedure *IsConsistent*$(\mathbf{E}, \mathcal{K})$ shown in Figure 4.1. *IsConsistent*$(\mathbf{E}, \mathcal{K})$ takes $\mathbf{E}$ and $\mathcal{K}$ as inputs and outputs a Boolean value `true` if $\mathbf{E}$ is consistent with $\mathcal{K}$; `false` otherwise. The procedure first checks if $|\mathbf{E}| = |\mathbf{Vars}(\mathbf{E}))|$. If $|\mathbf{E}| = |\mathbf{Vars}(\mathbf{E})|$, the procedure assumes that $\mathbf{E}$ is a self-contained structure and applies COA_{BGM} qualitatively on $\mathbf{E}$'s structure matrix to generate the graph $G(\mathbf{E})$. If $|\mathbf{E}| \neq |\mathbf{Vars}(\mathbf{E})|$, return `false`. For each partition $\mathbf{N}_k^p$ in $G(\mathbf{E})$, the procedure checks if the mapped structural equations $\widehat{\mathbf{C}_k^p}$ have corresponding explicit functional forms in $\mathcal{K}$, i.e., if there exists $\mathbf{N}_k^p \subseteq \mathbf{EfVars}(\widehat{\mathbf{C}_k^p})$. The procedure outputs `false` if for any $\mathbf{N}_k^p$, $\mathbf{N}_k^p \nsubseteq \mathbf{EfVars}(\widehat{\mathbf{C}_k^p})$; otherwise `true`. In order to assist modelers in hypothesizing causal relations for a mechanism whose reversibility is unknown, the procedure treats the mechanism with unknown reversibility as completely reversible. Notice that for those $\mathbf{E}$ containing strongly coupled components, those mechanisms in strongly-coupled components must have more than one effect variable. In other words, an irreversible mechanism cannot participate in a strongly coupled component. The time complexity of *IsConsistent*$(\mathbf{E}, \mathcal{K})$ is in worst-case poly-nominal because the procedure uses COA_{BGM} to generate the graph for consistency checking.

Example 4.1 Assume that the set of structural equations $\mathbf{E} = \{e_1, e_2, \ldots, e_8\}$ for the set of variables $\mathbf{V} = \{V_1, V_2, \ldots, V_8\}$ shown in Figure 2.3 is stored in a knowledge base $\mathcal{K}$ along with their explicit functional forms and the set of possible effect variables (See Table 4.1). In the knowledge base $\mathcal{K}$, e_6 and e_7 are irreversible where $\mathbf{EfVars}(e_6) = \{V_6\}$ and $\mathbf{EfVars}(e_7) = \{V_7\}$, e_4 and e_5 are completely reversible where $\mathbf{EfVars}(e_4) = \mathbf{Vars}(e_4)$ and $\mathbf{EfVars}(e_5) = \mathbf{Vars}(e_5)$, and e_8

Procedure *IsConsistent*$(\mathbf{E}, \mathcal{K})$

Input: A set of structural equations $\mathbf{E}$; A mechanism knowledge base $\mathcal{K}$.

Output: `true`, if $\mathbf{E}$ is consistent with $\mathcal{K}$; `false`, otherwise.

1. **if** $|\mathbf{E}| \neq |\mathbf{Vars}(\mathbf{E})|$, **return** `false`.
2. Apply COA_{BGM} on $\mathbf{E}$ and generate corresponding graph $G(\mathbf{E})$.
3. **for each** $\langle \mathbf{N}_k^p, \widehat{\mathbf{C}_k^p} \rangle$ in $G(\mathbf{E})$
4. **if** $\mathbf{N}_k^p \nsubseteq \mathbf{EfVars}(\widehat{\mathbf{C}_k^p})$, **return** `false`.
5. **end for each**
6. **return** `true`

Figure 4.1. Procedure *IsConsistent*$(\mathbf{E}, \mathcal{K})$ that tests if a set of structural equations $\mathbf{E}$ is consistent with a knowledge base $\mathcal{K}$.

Table 4.1. The fragment of a knowledge base $\mathcal{K}$ represents effect variables for the set of structural equations in Figure 2.3.

Structural Equation e	Effect Variables ($\mathbf{EfVars}(e)$)	Reversibility
$e_1(V_1) = 0$	$\{V_1\}$	*IR*
$e_2(V_2) = 0$	$\{V_2\}$	*IR*
$e_3(V_3) = 0$	$\{V_3\}$	*IR*
$e_4(V_2, V_4, V_5) = 0$	$\{V_2, V_4, V_5\}$	*CR*
$e_5(V_2, V_3, V_4, V_5) = 0$	$\{V_2, V_3, V_4, V_5\}$	*CR*
$e_6(V_3, V_5, V_6) = 0$	$\{V_6\}$	*IR*
$e_7(V_1, V_7) = 0$	$\{V_7\}$	*IR*
$e_8(V_1, V_7, V_8) = 0$	$\{V_1, V_8\}$	*PR*

is partially reversible where $\mathbf{EfVars}(e_8) = \{V_1, V_8\}$. Consequently, $\mathbf{E}$ is consistent with $\mathcal{K}$ since there exists a self-contained structure $\mathbf{F}$ of $\mathbf{E}$. However, if I have $\mathbf{EfVars}(e_7) = \{V_4\}$ instead of $\mathbf{EfVars}(e_7) = \{V_7\}$ in the knowledge base $\mathcal{K}$, then $\mathbf{E}$ is not consistent with $\mathcal{K}$ because there is no explicit functional form of e_7 that corresponds to $\mathbf{F}$ depicted by $G(\mathbf{E})$. □

4.3 Representation of Action

Given a causal model that describes a system of interest, we may easily hypothesize different manipulations on manipulable variables with the intention to influence the values of some target variables. For example, we can "lower interest rate" to increase "consumer spending." However, we may not know how other parts of the system may respond to these hypothetical manipulations. In other words, we know that our hypothetical manipulations will affect the variables of interest, which are usually the descendants of the manipulated variables in a causal graph, but we are not ceratin how other parts of the system will be disturbed by our hypothetical manipulations. Therefore, the process of policy making usually focuses on deliberating the unexpected effects of a manipulation. How should we represent an action in causal modeling to facilitate this deliberation?

Pearl [2000, pp. 225] suggested to use the notation $do(q)$, where q is a proposition, to denote an action, since people use phrases such as "reduce tax" in daily language to express actions. More precisely, an *atomic action*, denoted as $do(V = v)$ in [Pearl, 2000] and $manipulate(v)$ in [Spirtes *et al.*, 1993; Cooper, 1999], is invoked by an external force that manipulates on a variable V by imposing a probability distribution or holding it at a constant value, v, and replacing the causal mechanism, $V = f(\mathbf{Pa}(V))$, directly governing V in a causal model. The corresponding change in the causal graph is depicted as the arc-cutting operation by which all incoming arcs to the manipulated variable V are removed [Spirtes *et al.*, 1993; Pearl, 2000]. Notice that the implicit assumption behind the arc-cutting operation is that the manipulated variable is governed by an *irreversible* mechanism, i.e., only V can be an effect variable in mechanism $e(V, \mathbf{Pa}(V)) = 0$. In order to ensure that the manipulated causal model is self-contained, the irreversible mechanism, which governs the manipulated variable in the model before manipulation, has to be replaced. However, when the manipulated variable is governed by a *reversible* mechanism, the manipulated model derived from the arc-cutting operation may not be consistent with our conception of the manipulated system. I argue that an action in causal modeling should be defined at the level of mechanisms, not at the level of propositions.

In econometric literature (e.g., [Marschak, 1953; Simon, 1953; Wold, 1954; Strotz and Wold, 1960]), a system is represented as a SEM, a set of structural equations, and actions are modeled as "scraping invalid equations" and "replacing them by new ones". In STRIPS language [Fikes and

Nilsson, 1971], a situation is represented by a state, conjunctions of function-free ground literals (propositions), and actions are represented as *PRE*, *ADD*, and *DEL* lists which are conjunctions of literals. There is a clear analogy between these two modeling formalisms, where the effects of actions are modeled explicitly as adding or deleting fundamental building blocks, which are structural equations in SEM and propositions in STRIPS. I propose to explicitly translate the operations of "scraping invalid equations" as specifying equations in $\mathbf{E_{del}}$ list and "replacing them by new ones" as specifying equations in $\mathbf{E_{add}}$ and define an action operator as follows.

Definition 4.3 (action operator)

An action operator $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *represents an action on a system represented by a SEM* $\mathbf{E}$*, where* $\mathbf{E_{pre}}$ *is the precondition of the action, and* $\mathbf{E_{add}}$ *and* $\mathbf{E_{del}}$ *are the changes brought about by the action on* $\mathbf{E}$ *:*

1. $\mathbf{E_{pre}}$*: a set of conditions that must be true before the action can be applied to* $\mathbf{E}$*.*
2. $\mathbf{E_{add}}$*: a set of structural equations added to* $\mathbf{E}$*.*
3. $\mathbf{E_{del}}$*: a set of structural equations removed from* $\mathbf{E}$*.*

I remark that Definition 4.3 explicitly specifies the context and the effects of an action. This is consistent with our daily dialogue where we talk about an action and its possible effects under a certain context. For example, the phrase "reduce tax" is usually stated in an economic context with some expectations about how economic units would react.

Given an action $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ on a SEM $\mathbf{E}$, I define the manipulated model $\mathbf{E}_A$ as follows.

Definition 4.4 (manipulated model)

A manipulated model $\mathbf{E}_A$ *resulting from applying an action* $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *on a SEM* $\mathbf{E}$ *is a set of structural equations* $\mathbf{E}_A = \mathbf{E} \cup \mathbf{E_{add}} \setminus \mathbf{E_{del}}$*.*

Since an action $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ can be applied to a SEM $\mathbf{E}$ as long as the preconditions $\mathbf{E_{pre}}$ are satisfied, the manipulated model $\mathbf{E}_A$ is not necessary self-contained after the manipulation even though $\mathbf{E}$ is self-contained. $\mathbf{E}_A$ could be under-constrained or over-constrained, depending

on the structural equations specified in $\mathbf{E_{add}}$ and $\mathbf{E_{del}}$. To support the action deliberation in the following section, I define a *self-contained action* with respect to a self-contained model $\mathbf{E}$ as follows.

Definition 4.5 (self-contained action)

An action $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *on a system represented by a SEM* $\mathbf{E}$ *is* self-contained *if the manipulated model* $\mathbf{E}_A$ *is self-contained and indeed represents the manipulated system.*

Definition 4.5 states that applying a self-contained action on a self-contained model will result in a self-contained manipulated model. For example, an atomic action defined in [Spirtes *et al.*, 1993; Pearl, 2000] is considered a self-contained action in my definition.

4.4 Action Deliberation

Once I choose to represent an action explicitly including its effects and context, I shift the problem of predicting the effects of an action to the problem of finding which structural equations should be specified in $\mathbf{E_{add}}$ and $\mathbf{E_{del}}$. I call the process of deciding which structural equations should be in $\mathbf{E_{add}}$ and $\mathbf{E_{del}}$ *action deliberation*. In this section, I develop theorems to assist the process of deliberating about an atomic action. Given a SEM $\mathbf{E}$, I seek to answer two new types of queries (1) When making an endogenous variable exogenous, which structural equations are possibly invalidated and can be removed from the model? (2) Which manipulable variables may be manipulated in order to invalidate and, effectively, remove a structural equation from a model? Query (1) assists modelers in modeling the effects of an action considering the direct manipulation at hand; Query (2), on the other hand, assists modelers in identifying the set of possible manipulation alternatives. In order to assist action deliberation in systems with mixtures of mechanisms, I introduce the following definitions.

Definition 4.6 (atomic addition)

The $\mathbf{E_{add}}$ *of an action* $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *is atomic if it consists of only one structural equation in the form* $V = v$ *which assigns a variable* $V \in \mathbf{Vars}(\mathbf{E})$ *of a SEM* $\mathbf{E}$ *to a value* v*. I denote such atomic addition by* $\mathbf{E}_{add(V)}$*.*

Definition 4.7 (atomic deletion)

The $\mathbf{E_{del}}$ *of an action* $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *is* atomic *if it consists of only one structural equation* $e \in \mathbf{E}$ *of a SEM* $\mathbf{E}$*. I denote such atomic deletion by* $\mathbf{E}_{del(e)}$*.*

Definition 4.8 (atomic action)

An action $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *is atomic if it is a self-contained action where* $\mathbf{E_{add}} = \mathbf{E}_{add(V)}$ *and* $\mathbf{E_{del}} = \mathbf{E}_{del(e)}$*, and* $V \in \mathbf{Vars}(\mathbf{E})$ *and* $e \in \mathbf{E}$*.*

I will sometime use the notation $\mathbf{E}_V$ as a shorthand for a model resulting from applying an atomic action $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E}_{add(V)}, \mathbf{E}_{del(e)})$ on a model $\mathbf{E}$. I will assume that $\mathbf{E_{pre}}$ is satisfied in the rest of the discussion.

From Definition 2.10, I know that each variable V in a self-contained $\mathbf{E}$ can appear as an endogenous variable in only one $\widehat{\mathbf{C}_k^p}$. I define the *necessary structure* for V in $\mathbf{E}$ to support action deliberation.

Definition 4.9 (necessary structure)

Let $G(\mathbf{E})$ *be the causal graph generated by applying the causal ordering algorithm to a self-contained structure* $\mathbf{E}$*. Let* $V \in \mathbf{N}_k^p$ *and* $\mathbf{Anc}(\mathbf{N}_k^p)$ *be the ancestral set of* $\mathbf{N}_k^p$ *in* $G(\mathbf{E})$*. The* necessary structure *for* V*, denoted as* $\mathbf{E}_{ns(V)}$*, is the set of equations that map to* $\mathbf{N}_k^p \cup \mathbf{Anc}(\mathbf{N}_k^p)$ *by the causal ordering algorithm.*

It is easy to see that a necessary structure is self-contained. In other words, $\mathbf{E}_{ns(V_i)}$ consists of all equations in $\mathbf{E}$ that are necessary to determine V_i uniquely.

Example 4.2 In Example 2.3, for V_1, V_2, V_3, the necessary structures are $\mathbf{E}_{ns(V_1)} = \{e_1\}$, $\mathbf{E}_{ns(V_2)} = \{e_2\}$, and $\mathbf{E}_{ns(V_3)} = \{e_3\}$ respectively. The necessary structures for V_4 and V_5 are the same, $\mathbf{E}_{ns(V_4)} = \mathbf{E}_{ns(V_5)} = \{e_2, e_3, e_4, e_5\}$. For V_6, the necessary structure is $\mathbf{E}_{ns(V_6)} = \{e_2, e_3, e_4, e_5, e_6\}$. For V_7, the necessary structure is $\mathbf{E}_{ns(V_7)} = \{e_2, e_3, e_4, e_5, e_7\}$. For V_8, the necessary structure is $\mathbf{E}_{ns(V_8)} = \{e_1, e_2, e_3, e_4, e_5, e_7, e_8\}$. □

I define the set of *minimal over-constrained* equations to describe the situation where an atomic addition is made to a self-contained model.

Definition 4.10 (minimal over-constrained)

A set of over-constrained equations is minimal if itself does not contain any over-constrained proper subsets.

Lemma 4.1

Let $\mathbf{E}$ *be a self-contained structure and* $\mathbf{E}_{add(V)}$ *be an atomic addition where* $V \in \mathbf{EnVars}(\mathbf{E})$*. Let* $\mathbf{E}'_V = \mathbf{E}_{add(V)} \cup \mathbf{E}$*. The set of equations* $\mathbf{O}'_V = \mathbf{E}_{ns(V)} \cup \mathbf{E}_{add(V)}$ *is minimal over-constrained, where* $\mathbf{E}_{ns(V)}$ *is the necessary structure of* V *in* $\mathbf{E}$*.*

Proof: I first show that $\mathbf{O}'_V$ is over-constrained. We know that $|\mathbf{E}_{ns(V)}| = |\mathbf{Vars}(\mathbf{E}_{ns(V)})|$ because $\mathbf{E}_{ns(V)}$ is self-contained. Since $\mathbf{O}'_V = \mathbf{E}_{ns(V)} \cup \mathbf{E}_{add(V)}$ and the only variable in $\mathbf{E}_{add(V)}$ is also in $\mathbf{E}_{ns(V)}$, we have $|\mathbf{O}'_V| = |\mathbf{E}_{ns(V)}| + 1 > |\mathbf{Vars}(\mathbf{E}_{ns(V)})| = |\mathbf{Vars}(\mathbf{O}'_V)|$. Consequently, $\mathbf{O}'_V$ is not a structure by Definition 2.3. Therefore, $\mathbf{O}'_V$ is over-constrained. To show that $\mathbf{O}'_V$ is minimal, we need to show that any proper subset $\mathbf{P} \subset \mathbf{O}'_V$ is a structure. If $\mathbf{E}_{add(V)} \notin \mathbf{P}$, then $\mathbf{P} \subseteq \mathbf{E}_{ns(V)}$. Consequently, $\mathbf{P}$ is a structure since $\mathbf{E}_{ns(V)}$ is a self-contained structure. If $\mathbf{E}_{add(V)} \in \mathbf{P}$, consider the simple case where $\mathbf{P} = \mathbf{E}_{add(V)}$. Apparently, $\mathbf{P}$ is a self-contained structure. The only case left is $\mathbf{P} = \mathbf{Q} \cup \mathbf{E}_{add(V)}$, where $\mathbf{Q} \subset \mathbf{E}_{ns(V)}$. If $V \notin \mathbf{Vars}(Q)$, then $\mathbf{P}$ is a structure because $\mathbf{Q}$ is a structure itself and $\mathbf{E}_{add(V)}$ only determines the variable V. If $V \in \mathbf{Vars}(\mathbf{Q})$, we know that $|\mathbf{Q}| < |\mathbf{Vars}(\mathbf{Q})|$ because and $\mathbf{Q} \subset \mathbf{E}_{ns(V)}$ and Definitions 2.10 and 4.9 ensure that V appears as one of the undetermined variables in $\mathbf{Q}$. Therefore, $\mathbf{Q}$ is under-constrained. Consequently, $\mathbf{P}$ is a structure since $\mathbf{E}_{add(V)}$ only determines the variable V. We conclude that $\mathbf{O}'_V$ is minimal over-constrained. □

Lemma 4.1 states that an atomic addition makes a self-contained structure minimal over-constrained. Next, I prove Lemma 4.2 to identify the set of equations such that removing any one of them makes the set of minimal over-constrained equations self-contained again.

Lemma 4.2

Given $\mathbf{O}'_V$ *of* $\mathbf{E}'_V$*, deleting any equation* $e \in \mathbf{E}_{ns(V)}$ *makes* $\mathbf{O}_V = \mathbf{O}'_V \setminus \mathbf{E}_{del(e)}$ *self-contained and, consequently,* $\mathbf{E}_V = \mathbf{E}'_V \setminus \mathbf{E}_{del(e)}$ *self-contained.*

Proof: Since $\mathbf{O}_V \subset \mathbf{O}'_V$ and there is no proper subset of equations that is over-constrained in a set of minimal over-constrained equations by Definition 4.10, we know that $\mathbf{O}_V$ is a structure.

Now, we need to prove that $\mathbf{O}_V$ is self-contained, i.e., $|\mathbf{O}_V| = |\mathbf{Vars}(\mathbf{O}_V)|$. We first show that $|\mathbf{O}_V| = |\mathbf{E}_{ns(V)}|$. Let $\mathbf{R} = \mathbf{E}_{ns(V)} \setminus \mathbf{E}_{del(e)}$. We have $\mathbf{O}_V = \mathbf{R} \cup \mathbf{E}_{add(V)}$. Consequently, $|\mathbf{O}_V| = |\mathbf{R}| + 1 = |\mathbf{E}_{ns(V)}| - |\mathbf{E}_{del(e)}| + 1 = |\mathbf{E}_{ns(V)}| - 1 + 1 = |\mathbf{E}_{ns(V)}|$. Since we know that $|\mathbf{E}_{ns(V)}| = |\mathbf{Vars}(\mathbf{E}_{ns(V)})|$, all we need to prove now is $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$. Let $V \in \mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$ in $\mathbf{E}_{ns(V)}$. We consider cases where (a) $e \in \widehat{\mathbf{C}_k^p}$ and $\mathbf{C}_k^p$ is *not* strongly coupled in $\mathbf{E}_{ns(V)}$, (b) $e \in \widehat{\mathbf{C}_k^p}$ and $\mathbf{C}_k^p$ is strongly coupled in $\mathbf{E}_{ns(V)}$, and (c) $e \notin \widehat{\mathbf{C}_k^p}$.

In case (a), $e = \widehat{\mathbf{C}_k^p}$, since $\mathbf{C}_k^p$ is *not* strongly coupled. Therefore, $\mathbf{Vars}(e) = \mathbf{Vars}(\widehat{\mathbf{C}_k^p})$, i.e., $\mathbf{EnVars}(e) = \mathbf{EnVars}(\widehat{\mathbf{C}_k^p}) = \{V\}$ and $\mathbf{ExVars}(e) = \mathbf{ExVars}(\widehat{\mathbf{C}_k^p})$. By Definitions 2.3 and 4.9, and $\mathbf{R} = \mathbf{E}_{ns(V)} \setminus \mathbf{E}_{del(e)}$, we know that $\mathbf{ExVars}(e) \subseteq \mathbf{Vars}(\mathbf{R})$ and $V \notin \mathbf{Vars}(\mathbf{R})$. Therefore, $\mathbf{Vars}(\mathbf{R}) = \mathbf{Vars}(\mathbf{E}_{ns(V)}) \setminus \{V\}$. Since $\mathbf{O}_V = \mathbf{R} \cup \mathbf{E}_{add(v)}$, we know that $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{R}) \cup \mathbf{Vars}(\mathbf{E}_{add(V)}) = (\mathbf{Vars}(E_{ns(V)}) \setminus \{V\}) \cup \{V\} = \mathbf{Vars}(\mathbf{E}_{ns(V)})$. We have that $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$ in this case.

In case (b), since $\mathbf{C}_k^p$ is strongly coupled, we have $\mathbf{ExVars}(e) \subseteq \mathbf{ExVars}(\widehat{\mathbf{C}_k^p})$ and $\mathbf{EnVars}(e) = \mathbf{EnVars}(\widehat{\mathbf{C}_k^p})$. By Definitions 2.3 and 4.9, we know that $\mathbf{ExVars}(e) \subseteq \mathbf{Vars}(\mathbf{R})$ and $\mathbf{EnVars}(e) \subseteq \mathbf{Vars}(\mathbf{R})$ because there must exist at least one more equation in $\widehat{\mathbf{C}_k^p}$ that has the same set of endogenous variables as e when $\mathbf{C}_k^p$ is strongly coupled. We therefore have $\mathbf{Vars}(e) \subseteq \mathbf{Vars}(\mathbf{R})$. Consequently, $\mathbf{Vars}(\mathbf{E}_{ns(V)}) = \mathbf{Vars}(\mathbf{R}) \cup \mathbf{Vars}(e) = \mathbf{Vars}(\mathbf{R})$ and $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{R}) \cup \mathbf{Vars}(\mathbf{E}_{add(V)}) = \mathbf{Vars}(\mathbf{E}_{ns(V)}) \cup \mathbf{Vars}(\mathbf{E}_{add(V)}) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$. We prove that $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$ in this case.

In case (c), since $e \notin \widehat{\mathbf{C}_k^p}$, e must be in some $\widehat{\mathbf{C}_l^q}$ where $\widehat{\mathbf{C}_l^q} \subset \mathbf{E}_{ns(V)}$ and $q < p$. By Definitions 2.3 and 4.9, we know that $\mathbf{ExVars}(e) \subseteq \mathbf{ExVars}(\widehat{\mathbf{C}_l^q}) \subset \mathbf{Vars}(\widehat{\mathbf{C}^t})$ where $\widehat{\mathbf{C}^t} \subset \mathbf{R}$ and $t < q$, and $\mathbf{EnVars}(e) = \mathbf{EnVars}(\widehat{\mathbf{C}_l^q}) \subset \mathbf{Vars}(\widehat{\mathbf{C}^{t'}})$ where $\widehat{\mathbf{C}^{t'}} \subset \mathbf{R}$ and $q < t' \leq p$. Therefore, $\mathbf{Vars}(e) \subset \mathbf{Vars}(\mathbf{R})$. Then again $\mathbf{Vars}(\mathbf{E}_{ns(V)}) = \mathbf{Vars}(\mathbf{R}) \cup \mathbf{Vars}(e) = \mathbf{Vars}(\mathbf{R})$ and $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{R}) \cup \mathbf{Vars}(\mathbf{E}_{add(V)}) = \mathbf{Vars}(\mathbf{E}_{ns(V)}) \cup \mathbf{Vars}(\mathbf{E}_{add(V)}) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$. We prove that $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(\mathbf{E}_{ns(V)})$ in this case. We conclude that in all cases $|\mathbf{Vars}(\mathbf{O}_V)| = |\mathbf{O}_V|$. $\mathbf{O}_V$ is self-contained.

It is easy to see that $\mathbf{E}_V = \mathbf{E}'_V \setminus \mathbf{E}_{del(e)}$ is self-contained. In $\mathbf{E}$, all variables in $\mathbf{Vars}(\mathbf{E}_{ns(V)})$ are solved and may serve as exogenous variables to the rest of equations in $\mathbf{E}$. In $\mathbf{E}_V$, we see from above proof that $\mathbf{Vars}(\mathbf{O}_V) = \mathbf{Vars}(ns(V))$ and $\mathbf{O}_V$ is self-contained. Therefore, variables in $\mathbf{Vars}(\mathbf{E}_{ns(V)})$ are solved in $\mathbf{O}_V$ and may serve as exogenous variables to the rest of equations staying intact in $\mathbf{E}_V$. Therefore, $\mathbf{E}_V$ is self-contained. □

Corollary 4.1

Given $\mathbf{E}'_V = \mathbf{E}_{add(V)} \cup \mathbf{E}$, $\mathbf{E}'_V$ *will remain over-constrained if none of equations* $e \in \mathbf{O}'_V$ *is removed.*

Example 4.3 Consider the self-contained structure $\mathbf{E}$ in Figure 2.3. If we consider an atomic addition on variable V_7, i.e., $\mathbf{E}_{add(V_7)}$, the resulting set of equations $\mathbf{E}'_{V_7} = \mathbf{E} \cup \mathbf{E}_{add(V_7)}$ becomes over-constrained. From Lemma 4.1, we know that the set of equations $\mathbf{O}'_{V_7} = \{e_2, e_3, e_4, e_5, e_7, \mathbf{E}_{add(V_7)}\}$ is minimal over-constrained. From Lemma 4.2, we know that removing any equation $e \in \{e_2, e_3, e_4, e_5, e_7\}$ makes the remaining set of equations $\mathbf{E}_{V_7} = \mathbf{E}'_{V_7} \setminus \mathbf{E}_{del(e)}$ a self-contained structure. If we instead remove e_6, the set of equations $\mathbf{E}'_{V_7} \setminus \mathbf{E}_{del(e_6)}$ remains over-constrained according to Corollary 4.1.

□

Notice that Lemmas 4.1 and 4.2 hold for a sets of structural equations. As stated in Section 4.2, when assisting changes in structure on systems containing reversible mechanisms, we need to verify if a manipulated model is consistent with the knowledge base $\mathcal{K}$. Therefore, in order to deliberate about an atomic action, we need to verify that the manipulated set of structural equations is consistent with $\mathcal{K}$. In general, we can simply enumerate each structural equation $e \in \mathbf{E}_{ns(V)}$ and then use the procedure *IsConsistent*$(\mathbf{E}_V, \mathcal{K})$ outlined in Figure 4.1 to check if the manipulated model $\mathbf{E}_V$ is consistent with $\mathcal{K}$.

However, please observe that the irreversibility of mechanisms allows us to find the set of possible atomic deletions *locally*. Consider an atomic addition $\mathbf{E}_{add(V)}$ on a SEM $\mathbf{E} = \{e_1, e_2, \ldots, e_m\}$ and $V \in \mathbf{EnVars}(\mathbf{E})$. When all mechanisms governing $\mathbf{EnVars}(\mathbf{E}_{ns(V)})$ in $\mathbf{E}_{ns(V)}$ are completely reversible or unknown, we may remove any one of the mechanisms in $\mathbf{E}_{ns(V)}$ to have a manipulated SEM that is consistent with $\mathcal{K}$. When V is directly governed by an irreversible mechanism e, we have to remove e since V cannot be determined by $\mathbf{E}_{add(V)}$ and e simultaneously in a manipulated model. In other words, the irreversibility of the mechanism which governs the manipulated variable reduces the set of possible atomic deletions from $\mathbf{E}_{ns(V)}$ to e. Based on the above observations, we see that the propagation of the impacts of an atomic addition on a SEM is blocked by irreversible mechanisms. Now, I can prove Theorem 4.1 to answer Query (1): When making an endogenous variable exogenous, which structural equations are possibly invalidated and can be removed from the model?

Theorem 4.1 (possible atomic deletions)

Let $\mathbf{E}_{add(V)}$ *be an atomic addition on a SEM* $\mathbf{E}$ *consistent with a knowledge base* $\mathcal{K}$*, where* $V \in \mathbf{EnVars}(\mathbf{E})$*. Then, there exists a non-empty set of possible atomic deletions* $\mathbf{D} \subseteq \mathbf{E}_{ns(V)}$ *such that deleting any structural equation* $d \in \mathbf{D}$ *derives the manipulated SEM,* $\mathbf{E}_V = \mathbf{E} \cup \mathbf{E}_{add(V)} \setminus \mathbf{E}_{del(d)}$*, that is consistent with* $\mathcal{K}$*.*

Proof: From Lemmas 4.1 and 4.2, we know that removing any equation $e \in \mathbf{E}_{ns(V)}$ makes $\mathbf{E}_V = \mathbf{E} \cup \mathbf{E}_{add(V)} \setminus \mathbf{E}_{del(e)}$ self-contained. Now, we need to check whether the manipulated $\mathbf{E}_V$ is consistent with $\mathcal{K}$. Let $G(\mathbf{E})$ be the causal graph of $\mathbf{E}$. Let $G(\mathbf{E}_{ns(V)})$ be the subgraph of $G(\mathbf{E})$ induced by the set of nodes representing variables in $\mathbf{Vars}(\mathbf{E}_{ns(V)})$. For each $\mathbf{N}_k^p$ in $G(\mathbf{E}_{ns(V)})$, if $\widehat{\mathbf{C}_k^p}$, where $\langle \widehat{\mathbf{C}_k^p}, \mathbf{N}_k^p \rangle$ in $G(\mathbf{E})$, is irreversible, cut all incoming arcs to $\mathbf{N}_k^p$ in $G(\mathbf{E}_{ns(V)})$ and derive the graph $G'(\mathbf{E}_{ns(V)})$. Let $\mathbf{D}$ be the set of equations mapped to nodes $\mathbf{Anc}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p$, where $V \in \mathbf{N}_k^p$, in $G'(\mathbf{E}_{ns(V)})$. For each $d \in \mathbf{D}$, let $\mathbf{E}_V = \mathbf{E} \cup \mathbf{E}_{add(V)} \setminus \mathbf{E}_{del(d)}$. If $\mathit{IsConsistent}(\mathbf{E}_V, \mathcal{K})$ is false, then remove d from $\mathbf{D}$. Consequently, we construct the set of possible atomic deletions $\mathbf{D}$. Notice that $\mathbf{D}$ is complete since Corollary 4.1 ensures that $\mathbf{D} \subseteq \mathbf{E}_{ns(v)}$ and we only use the irreversibility of mechanisms to avoid unnecessary checking. Also notice that $\mathbf{D}$ is not empty since we can at least remove the structural equation directly governing V in $\mathbf{E}$. Therefore, we prove by construction that there exists a nonempty set of possible atomic deletions $\mathbf{D} \subseteq \mathbf{E}_{ns(V)}$ such that deleting any one of mechanisms $d \in \mathbf{D}$ derives a manipulated $\mathbf{E}_V$ that is consistent with $\mathcal{K}$. □

In Figure 4.2, I outline the procedure *FindAtomicDeletions* that based on Theorem 4.1 identifies the set of possible atomic deletions. The procedure takes a self-contained structure $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and an atomic addition $\mathbf{E}_{add(V)}$ on $V \in \mathbf{Vars}(\mathbf{E})$ as inputs and generate the set of possible atomic deletions $\mathbf{D}$ as output. *FindAtomicDeletions* first computes the causal graph $G(\mathbf{E})$ using COA_{BGM} and identifies the subgraph $G(\mathbf{E}_{ns(V)})$ induced by $\mathbf{Vars}(\mathbf{E}_{ns(V)})$. Next, *FindAtomicDeletions* modifies $G(\mathbf{E}_{ns(V)})$ according to the principle that the propagation of the impact of an atomic addition is blocked by irreversible mechanisms. *FindAtomicDeletions* then enumerates each equation mapped to each node in the modified $G(\mathbf{E}_{ns(V)})$ as the atomic deletion d and checks if $\mathbf{E}_V = \mathbf{E} \cup \mathbf{E}_{add(V)} \setminus \mathbf{E}_{del(d)}$ is consistent with $\mathcal{K}$. *FindAtomicDeletions* then returns all consistent atomic deletions as $\mathbf{D}$. The complexity of *FindAtomicDeletions* is in worst case $O(n^{3/2}a)$ where n is the number of equations of $\mathbf{E}$ and a is the number of arcs in the bipartite

graph representation of $\mathbf{E}$, because the procedure *IsConsistent* is used in Line 12.

Procedure *FindAtomicDeletions*$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{add(V)})$

Input: A self-contained structure $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and an atomic addition $\mathbf{E}_{add(V)}$ on $V \in \mathbf{Vars}(\mathbf{E})$.

Output: The set of possible atomic deletions $\mathbf{D}$.

1. Apply COA_{BGM} on $\mathbf{E}$ and generate the corresponding graph $G(\mathbf{E})$.
2. Identify $\mathbf{Anc}(N_V)$ and induce the subgraph $G(\mathbf{E}_{ns(V)})$ of $G(\mathbf{E})$.
3. **for each** $\mathbf{N}_k^p$ in $G(\mathbf{E}_{ns(V)})$
4. **if** $\mathbf{N}_k^p$'s mapping $\widehat{\mathbf{C}_k^p}$ is irreversible in $\mathcal{K}$
5. Remove all incoming arcs to $\mathbf{N}_k^p$ in $G(\mathbf{E}_{ns(V)})$.
6. **end if**
7. **end for each**
8. Identify $\mathbf{Anc}(N_V) \cup N_V$ in $G(\mathbf{E}_{ns(V)})$.
9. Identify the set of equations mapped to $\mathbf{Anc}(N_V) \cup N_V$ as $\mathbf{D}$.
10. **for each** $d \in \mathbf{D}$
11. $\mathbf{E}_V := \mathbf{E} \cup \mathbf{E}_{add(V)} \setminus \mathbf{E}_{del(d)}$.
12. **if** *IsConsistent*$(\mathbf{E}_V, \mathcal{K})$ == `false`
13. $\mathbf{D} := \mathbf{D} \setminus \mathbf{E}_{del(d)}$.
14. **end if**
15. **end for each**
16. **return D**.

Figure 4.2. Procedure for finding the set of possible atomic deletions defined in Theorem 4.1.

Considering a completely reversible system, a manipulation usually results in the change of an operational context as in from the operation of driving uphill to the operation of driving downhill in the power train system described in Section 4.1. We normally remove one of the exogenous equations, when we manipulate on a system to change its operational contexts. Since all mechanisms in the system are still working, they are just possibly operating in different causal directions with respect to different operational contexts. However, if we remove a mechanism that was governing

an endogenous variable, it means that the linkage among the set of variables is invalid in the manipulated system. For example, transmission between the engine and the wheels may be broken. Consequently, the link between engine and wheel is no longer valid. Therefore, I suggest modelers to use different enumeration orders to inspect the set of possible atomic deletions in different applications.

Example 4.4 Consider the set of structural equations in Figure 2.3 and its reversibility assumed in Example 4.1. If we manipulate variable V_8, i.e., $\mathbf{E}_{add(V_8)}$, the set of possible atomic deletions is $\{e_1, e_8\}$ according to Theorem 4.1. Notice that the irreversibility of mechanisms allows us to find the set of possible atomic deletions in $\{e_1, e_7, e_8\}$ instead of $\mathbf{E}_{ns(V_8)}$. Moreover, $\mathbf{E}_{V_8} = \mathbf{E} \cup \mathbf{E}_{add(V_8)} \setminus \mathbf{E}_{del(e_7)}$ is not consistent with $\mathcal{K}$ since V_7 cannot be an effect variable in e_8 according to the reversibility of e_8 in the knowledge base $\mathcal{K}$. However, if we choose to remove e_1, i.e., $\mathbf{E}_{del(e_1)}$, the manipulated model is shown in Figure 4.3. □

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
$\mathbf{E}_{add(V_8)}$	0	0	0	0	0	0	0	x
e_2	0	x	0	0	0	0	0	0
e_3	0	0	x	0	0	0	0	0
e_4	0	x	0	x	x	0	0	0
e_5	0	x	x	x	x	0	0	0
e_6	0	0	x	0	x	x	0	0
e_7	0	0	0	x	0	0	x	0
e_8	x	0	0	0	0	0	x	x

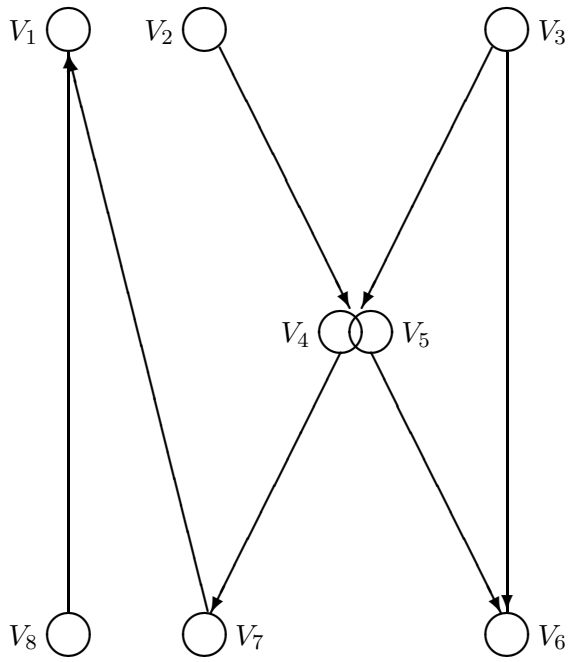


Figure 4.3. The structure matrix and its corresponding graph after the atomic action $\mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E}_{add(V_8)}, \mathbf{E}_{del(e_1)})$ on $\mathbf{E}$ in Figure 2.3.

The dual theorem to Theorem 4.1 is to identify the set of possible atomic additions given an atomic deletion, which answers Query (2): Which variables may be manipulated in order to invalidate and, effectively, remove a structural equation from a model?

Theorem 4.2 (possible atomic additions)

Let $\mathbf{E}_{del(e)}$ *be an atomic deletion on a SEM* $\mathbf{E}$ *consistent with a knowledge base* $\mathcal{K}$*, where* $e \in \mathbf{E}$*. Let* $\langle \mathbf{N}_k^p, \widehat{\mathbf{C}_k^p} \rangle$ *be the mapping in* $G(\mathbf{E})$*, where* $e \in \widehat{\mathbf{C}_k^p}$ *and* $G(\mathbf{E})$ *is the causal graph of* $\mathbf{E}$*. Let* $\mathbf{Des}(\mathbf{N}_k^p)$ *be the descendants of* $\mathbf{N}_k^p$ *in* $G(E)$*. Then, there exists a nonempty set of variables* $\mathbf{A} \subseteq (\mathbf{Des}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p)$ *such that manipulating any variable* $A \in \mathbf{A}$ *derives the manipulated SEM* $\mathbf{E}_A = \mathbf{E} \cup E_{add(A)} \setminus \mathbf{E}_{del(e)}$ *consistent with* $\mathcal{K}$*. The set of structural equations* $\bigcup_{A \in \mathbf{A}} \mathbf{E}_{add(A)}$ *is called the set of possible atomic additions.*

Proof: From Theorem 4.1, we know that in order to have $\mathbf{E}_A = \mathbf{E} \cup \mathbf{E}_{add(A)} \setminus \mathbf{E}_{del(e)}$ consistent with $\mathcal{K}$, we must have $e \in \mathbf{E}_{ns(A)}$. Consequently, A must be in $(\mathbf{Des}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p)$. Again, the irreversibility of mechanisms allows me to inspect smaller sets of variables. For each variable V in $(\mathbf{Des}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p)$, if V is governed by an irreversible mechanism, cut all arcs into V and derive the graph $G'(\mathbf{E})$. Let $\mathbf{A} = \mathbf{Des}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p$ in $G'(\mathbf{E})$. For each $A \in \mathbf{A}$, if $\mathbf{E}_A = \mathbf{E} \cup \mathbf{E}_{add(A)} \setminus \mathbf{E}_{del(e)}$ is not consistent with $\mathcal{K}$, we remove A from $\mathbf{A}$. Consequently, we construct the set of possible atomic additions $\bigcup_{A \in \mathbf{A}} \mathbf{E}_{add(A)}$. Notice that $\mathbf{A}$ is complete since Theorem 4.1 ensures that A is in $(\mathbf{Des}(\mathbf{N}_k^p) \cup \mathbf{N}_k^p)$ and we only use the irreversibility of mechanism to avoid unnecessary checking. Also notice that $\mathbf{A}$ is not empty since we can at least manipulate the variables governed by the mechanism e in $\mathbf{E}$. Therefore, we prove the theorem by construction. □

Theorem 4.2 states that in order to invalidate a structural equation in a model, we may manipulate on one of the variables in the set of possible atomic additions. Similarly, Theorem 4.2 assists modelers in finding the set of possible atomic additions locally. I outline the procedure *FindAtomicAdditions* in Figure 4.4. The complexity of *FindAtomicAdditions* is also in worst case $O(n^{3/2}a)$, where n is the number of equations of $\mathbf{E}$ and a is the number of arcs in the bipartite graph representation of $\mathbf{E}$.

Example 4.5 Consider the set of structural equations in Figure 2.3 and its reversibility assumed in Example 4.1. The set of possible atomic additions for structural equation e_4, $\mathbf{E}_{del(e_4)}$, is $\{V_4, V_5\}$

Procedure *FindAtomicAdditions*$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{del(e)})$

Input: A self-contained structure $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and an atomic deletion $\mathbf{E}_{del(e)}$ on $e \in \mathbf{E}$.

Output: The set of possible atomic additions $\mathbf{A}$.

1. Apply COA_{BGM} on $\mathbf{E}$ and generate the corresponding graph $G(\mathbf{E})$.
2. Identify $\mathbf{Des}(\mathbf{N}_l^q)$ in $G(\mathbf{E})$, where $\mathbf{N}_l^q$ is mapped with e.
3. **for each** $\mathbf{N}_k^p$ in $\mathbf{Des}(\mathbf{N}_l^q) \cup \mathbf{N}_l^q$
4. **if** $\mathbf{N}_k^p$'s mapping $\widehat{\mathbf{C}_k^p}$ is irreversible in $\mathcal{K}$
5. Remove all incoming arcs to $\mathbf{N}_k^p$ in $G(\mathbf{E})$.
6. **end if**
7. **end for each**
8. Let $\mathbf{A}$ be $\mathbf{Des}(\mathbf{N}_l^q) \cup \mathbf{N}_l^q$ in $G(\mathbf{E}_{ns(V)})$.
9. **for each** $A \in \mathbf{A}$
10. $\mathbf{E}_A := \mathbf{E} \cup \mathbf{E}_{add(A)} \setminus \mathbf{E}_{del(e)}$.
11. **if** *IsConsistent*$(\mathbf{E}_A, \mathcal{K})$ == `false`
12. $\mathbf{A} := \mathbf{A} \setminus \{A\}$.
13. **end if**
14. **end for each**
15. **return A**.

Figure 4.4. Procedure for finding the set of possible atomic additions defined in Theorem 4.2.

according to Theorem 4.2. If we choose to manipulate on V_5, $\mathbf{E}_{add(V_5)}$, the manipulated model and its causal graph are shown in Figure 4.5. □

4.5 Changes in Structure in *ImaGeNIe*

The tree view of mechanism libraries in *ImaGeNIe* provides graphical interface for users to interact with mechanism knowledge base. Users can create a mechanism in a mechanism library and organize the domain knowledge hierarchically into subsystems in the domain. To specify the infor-

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
e_1	x	0	0	0	0	0	0	0
e_2	0	x	0	0	0	0	0	0
e_3	0	0	x	0	0	0	0	0
$\mathbf{E}_{add(V_5)}$	0	0	0	0	x	0	0	0
e_5	0	x	x	x	x	0	0	0
e_6	0	0	x	0	x	x	0	0
e_7	0	0	0	x	0	0	x	0
e_8	x	0	0	0	0	0	x	x

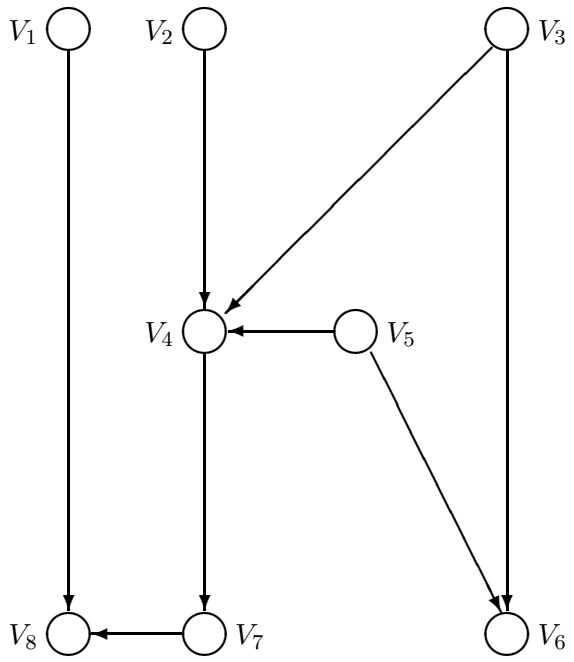


Figure 4.5. The structure matrix and its corresponding graph after the atomic action $\mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E}_{add(V_5)}, \mathbf{E}_{del(e_4)})$.

mation needed for changes in structure, users can open the property page of a mechanism to specify the properties associated with variables in a mechanism, namely manipulability, observability, and effectiveness. Please see Figure 4.6 for an example property page of a mechanism.

FindAtomicDeletions$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{add(V)})$ and *FindAtomicAdditions*$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{del(e)})$ have been implemented in *ImaGeNIe* to support atomic action deliberation for changes in structure. Users can select "Control Value" from the context menu of an endogenous variable to invoke the dialogue for specifying a value for an endogenous variable. If the system is self-contained, the "Release equation" drop-down list will contain equations in the set of possible atomic deletions produced by *FindAtomicDeletions*$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{add(V)})$. Users can select "Release Value" from the context menu of an exogenous variable to invoke the dialogue for releasing a value for an exogenous variable. If the system is self-contained, the "Add equation" drop-down list will contain equations in the set of possible atomic additions produced by *FindAtomicAdditions*$(\mathbf{E}, \mathcal{K}, \mathbf{E}_{del(e)})$.

Let's continue our example model building session in Section 3.7. After inspecting the current self-contained model, Tom would like to analyze the model under the condition that the average

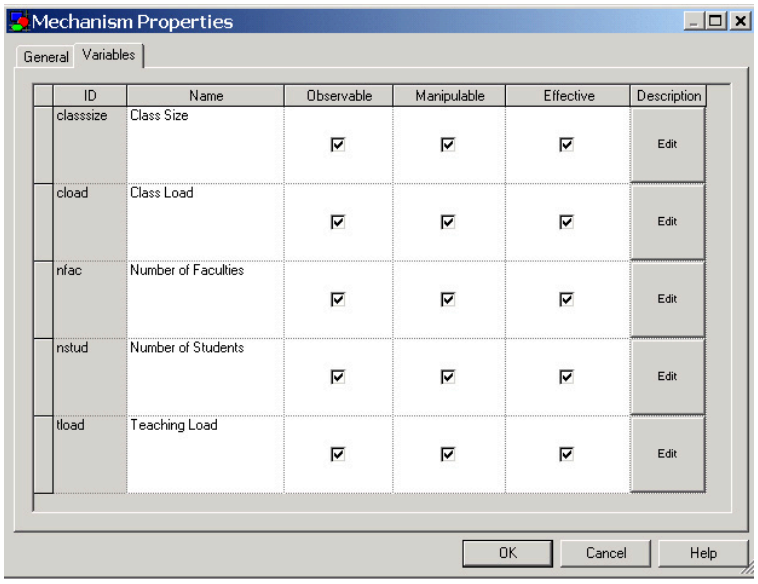


Figure 4.6. The property page for specifying manipulability, observability, and effectiveness for variables in the mechanism governing variables class size, number of faculty, number of student, class load, and teaching load..

class size is fixed at 15 students per class. He makes the variable *classsize* exogenous by specifying an exogenous equation equation as $classsize = 15$. Consequently, the original self-contained model will become over-constrained. *ImaGeNIe* will ask the model builder to release one of the equations. Suppose that he chooses to release the exogenous equation for the variable teaching load (*tload*). The resulting graph generated by the causal ordering is shown in Figure 4.7.

Now, he can read off another set of causal relations that correspond to the change of the system that he intends to model.

- *Teaching load* is determined by the *number of students*, the *number of faculty*, *class load* and *class size*.
- *Faculty salary* is determined by the *number of students*, the *number of faculty*, *tuition fee*, *other income*, and *overhead*.
- *Student-teacher ratio* is determined by the *number of students* and the *number of faculty*.

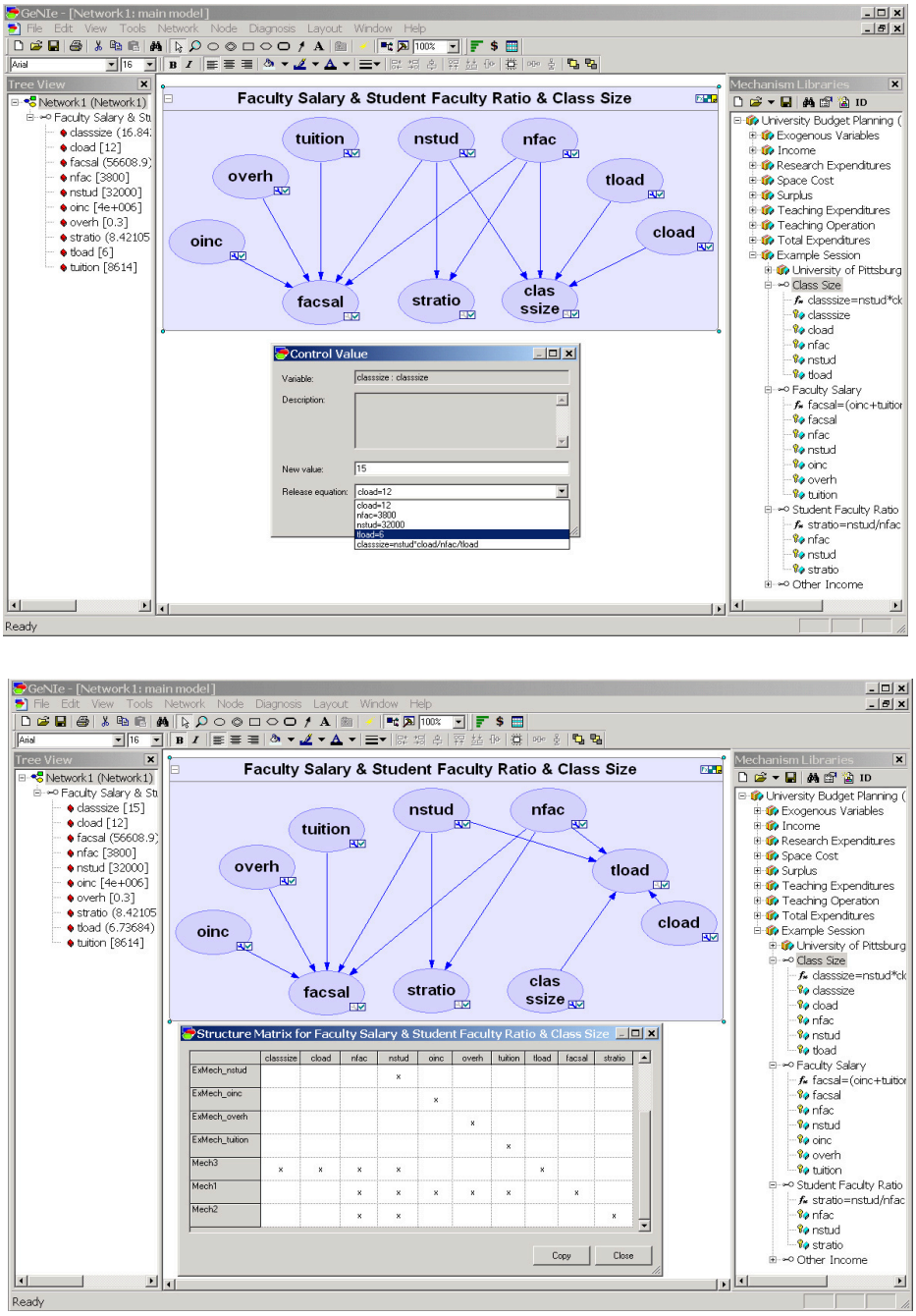


Figure 4.7. Model builder adds a exogenous equation for *class size* (*classsize*) and selects the exogenous equation for *teaching load* (*tload*) to be released (top), and the causal ordering generates the corresponding graph and its structure matrix (bottom).

4.6 Discussion

In this chapter, I have formalized the representation for the reversibility of a mechanism to support modeling of changes in structure. I have defined the reversibility of a mechanism semantically on the set of possible effect variables. This definition allows one to extend the concept of reversible mechanisms from traditional mechanical and physical systems to other systems. I have further drawn the analogy between the action represented in SEM and STRIPS languages to argue that the context and the effects of an action should be represented explicitly in causal modeling. My formalization allows one to answer two new types of queries: (1) When manipulating a causal model, which mechanisms are possibly invalidated and can be removed from the model? (2) Which variables may be manipulated in order to invalidate and, effectively, remove a mechanism from a model? In practical applications, it may be desirable to further encode domain knowledge, such as what is the cost of a manipulation, along with each mechanism.

Chapter 5
Search for Opportunities

The previous chapter discusses how to support atomic action deliberation for changes in structure given an atomic addition or an atomic deletion. In this chapter, I propose to address a decision scenario in which none of atomic addition or atomic deletion is given but a causal model and a decision objective are given. This decision scenario happens when a decision maker who is confronted with a complex system does not know which variables to best manipulate to achieve a desired objective. While algorithms for influence diagrams allow for computing the optimal setting for decision variables, they offer no guidance in generation of policy alternatives, arguably a critical stage of decision making. I introduce the problem of search for opportunities, which amounts to both identifying the set of policy variables and computing their optimal setting for a given decision objective. Search for opportunities is built on the value of intervention computation in causal models.

Section 5.1 introduces my motivation of addressing the problem of search for opportunities. Section 5.2 gives an overview of probabilistic causal models. Section 5.3 introduces augmented models for describing decision problems at hand. Section 5.4 discusses the concept of value of intervention. Section 5.5 shows the use of the value of intervention for solving the problem of search for opportunities. Section 5.6 presents the method of augmenting a model for non-intervening actions and the computation of value of observation. Section 5.7 discusses search for opportunities with a sequence of non-intervening and intervening actions on systems containing only irreversible mechanisms. Section 5.8 discusses search for opportunities with a sequence of non-intervening and intervening actions on systems containing reversible mechanisms.

5.1 Introduction

Influence diagrams [Howard and Matheson, 1981] are popular tools for representing decision problems under uncertainty and identifying optimal strategies. The key problem with using influence diagrams for decision support under uncertainty is that we need to specify beforehand all

decision alternatives and their consequences explicitly. In complex systems, this may result in a cumbersome, if not totally unmanageable, modeling process. Ideally, a modeling language should support the prediction of the effects of actions that were not considered in the construction of model [Druzdzel and Simon, 1993; Pearl, 1988; 2000]. This allows us to search for the best actions to be taken to achieve a set of objectives, a concept that I refer to as *search for opportunities*.

The problem of search for opportunities is related to the problem of *information gathering* [Russell and Norvig, 1995]. In information gathering, a decision maker tries to decide which information to acquire to reduce the uncertainty over a model, and consequently to improve the quality of the decision at hand. The means for acquiring information is constrained to observations that modify the decision maker's belief over the states of a system. Search for opportunities, in contrast, seeks to apply intervening actions that alter the trajectory of the system toward those outcomes that are preferable to the decision maker.

In decision analysis, the primary tool for information gathering is *value of information* [Howard, 1966] (also called *value of observation* [Savage, 1972]). Value of information is defined as the upper bound on what a decision maker should be willing to pay in employing a clairvoyant to reveal the outcome of a chance variable. Similarly, the concept of *value of control* has been introduced and defined as the upper bound on what a decision maker should be willing to pay a wizard for setting a chance node into a preferred state. Both value of information and value of control are defined with respect to a decision problem [Howard, 1971; Matheson, 1990]. However, to my knowledge, the value of control computation has only been applied to chance nodes with no predecessors in influence diagrams encoded in Howard canonical form [Howard, 1971; Matheson, 1990]. Since influence diagrams may describe probabilistic rather than causal relations, there is no guarantee that converting a chance node with predecessors (or a chance node without predecessors in diagrams that are not in Howard canonical form) into a decision node will correctly model the effects of control. Even influence diagrams in canonical form [Heckerman and Shachter, 1995], an extension of Howard canonical form that supports causal reasoning, require modelers to make causal assertions ("responsiveness") about chance variables with respect to a set of decisions, along with probabilistic assessments over *mapping* variables. This approach needs a combinatorial number of tests of the form: Is a node X_i responsive to a set of decisions D? This number can be very large if the tests are to be applied to all potential sets of decisions.

In this chapter, I discuss the problem of search for opportunities, where a decision maker seeks creative decision options in order to achieve a given objective. The basis of the search is a causal model of the system that is subject of the decision. Causal models based on structural equations support causal reasoning and, in particular, prediction of the effects of actions [Pearl, 2000; Spirtes *et al.*, 2000]. A causal model consists of a self-contained set of simultaneous structural equations, each of which represents a causal mechanism active in the modeled system. Causal models support prediction of the effects of actions by replacing those mechanisms that are impacted by actions with new mechanisms, possibly not contemplated during model's construction, and leaving the rest intact. The problem of search for opportunities, in this formulation, amounts to searching for variables that were not originally contemplated as decision variables, but were a priori specified as subject to potential manipulation, and intervening into mechanisms governing these variables in order to affect the outcomes. Therefore, search for opportunities leads to discovery of *novel* actions to achieve decision objectives.

To address the problem of search for opportunities, I introduce the concept of *value of intervention*. The value of intervention, related to the value of control, arises from considering jointly the economic factors and effects of actions in causal models. It can be considered a generalization of the value of control since the intervention operates at the level of mechanisms in causal models, but the control operates at the level of variables in influence diagrams. The value of intervention computation is also applicable to influence diagrams in canonical form [Heckerman and Shachter, 1995], but not to influence diagrams that do not represent causal relations.

5.2 Probabilistic Causal Models

Bayesian networks [Pearl, 1988] and influence diagrams [Howard and Matheson, 1981] are popular tools for reasoning and decision making under uncertainty. Both formalisms were originally developed for modeling probabilistic relations in the world. Druzdzel and Simon [1993] established the link between Bayesian networks and causal models. They showed that for the probability distribution encoded in a Bayesian network, there exists a structural equation model that generates the encoded distribution. Therefore, if each group of nodes, consisting of a node and its predecessors, in a Bayesian network represents a causal mechanism, then the Bayesian network can be interpreted

causally. Pearl [2000] presents a detail account of causal reasoning in probabilistic causal models based on structural equations. I briefly summarize the definition of probabilistic causal models based on structural equations as follows.

A *causal model* $M = \langle \mathbf{X}, \mathbf{E} \rangle$ consists of a self-contained set of simultaneous structural equations $\mathbf{E}$ over a set of variables $\mathbf{X} \equiv \mathbf{Vars}(\mathbf{E})$. The set of variables $\mathbf{X}$ can be partitioned into two disjoint sets $\mathbf{U} \equiv \mathbf{ExVars}(\mathbf{E})$ and $\mathbf{V} \equiv \mathbf{EnVars}(\mathbf{E})$ of exogenous and endogenous variables respectively. A causal model is sometimes denoted as $M = \langle \mathbf{U}, \mathbf{V}, \mathbf{E} \rangle$. Let $\mathsf{D}(X_i)$ be the domain of a variable X_i, and $\mathsf{D}(\mathbf{X}) = \mathsf{D}(X_1) \times \ldots \times \mathsf{D}(X_n)$ be the domain of the set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. Given $\mathbf{u} \in \mathsf{D}(\mathbf{U})$, the solutions for endogenous variables $\mathbf{Y} \subseteq \mathbf{V}$, denoted as $\mathbf{Y}_M(\mathbf{u})$ or $\mathbf{Y}(\mathbf{u})$, in a causal model M can always be determined uniquely. The pair $\langle M, \mathbf{u} \rangle$ is called a *causal world*, or simply world. Given a probability distribution $\Pr(\mathbf{u})$ defined over $\mathsf{D}(\mathbf{U})$, the pair $\langle M, \Pr(\mathbf{u}) \rangle$ is called a *probabilistic* causal model where for each $Y \in \mathbf{V}$, $\Pr(y) \equiv \Pr(Y = y) \triangleq \sum_{\{\mathbf{u} | Y(\mathbf{u}) = y\}} \Pr(\mathbf{u})$. A causal model M is *recursive* if the associated $G(M)$ is a directed acyclic graph, where each node corresponds to a variable, and each family (a node with its parents in $G(M)$) to a structural equation [Druzdzel and Simon, 1993]. In other words, each structural equation $e(X_1, \ldots, X_n) = 0$ is expressed in its explicit functional form $X_i = f_{X_i}(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ and is depicted graphically as a family with arcs from nodes representing arguments of f_{X_i} (i.e., $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n$) to X_i. For the rest of this chapter, the term "causal model" refers to a recursive probabilistic causal model, in which each equation is indexed by the dependent variable of its explicit form.

Example 5.1 Consider a model for the operational status of a command center (CC). CC depends on the status of communications (C) and radar (R). Radar depends on the antenna structure (A) and the power supplied by the generator (G). Communications relies on the power supplied by generator. The generator relies on fuel supply (F) to generate power. Each of the variables has state *operational* or *damaged*. We assume that for each of the above relations there is a corresponding exogenous variable, denoted as U_{cc}, U_r, U_c, and U_g, that summarizes the factors outside the models for each relation. F and A are themselves exogenous variables. We assume that all exogenous variables $U = \{U_{cc}, U_r, U_c, U_g, F, A\}$ are independent. The set of structural equations, representing the domain of our interest, and its corresponding causal graph are shown in Figure 5.1. We have included an explicit graphical representation of variables U_{cc}, U_r, U_c, U_g for the sake of clarity of

explanation. In practice, these variables are modeled implicitly by error terms in the corresponding equations and we will omit them in the sequel of this chapter for the sake of clarity. □

$$
\begin{aligned}
e_{U_g}(U_g) &= 0 \\
e_{U_c}(U_c) &= 0 \\
e_{U_r}(U_r) &= 0 \\
e_{U_{cc}}(U_{cc}) &= 0 \\
e_F(F) &= 0 \\
e_A(A) &= 0 \\
e_G(G, F, U_g) &= 0 \\
e_C(C, G, U_c) &= 0 \\
e_R(R, A, G, U_r) &= 0 \\
e_{CC}(CC, G, C, U_{cc}) &= 0
\end{aligned}
$$

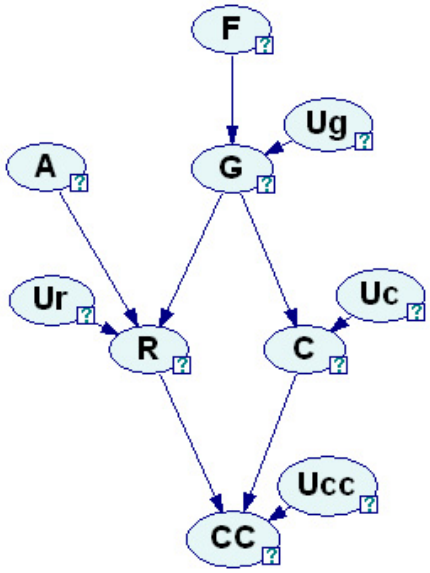


Figure 5.1. Causal model and its corresponding causal graph for modeling the operational status of a command center.

5.2.1 Recursive Actions

In general, an action on a recursive model may result in non-recursive models. Consider a simple recursive causal model $M = \langle \mathbf{X}, \mathbf{E} \rangle$ where $\mathbf{X} = \{X_1, X_2\}$ and $\mathbf{E} = \{e_{X_1}(X_1) = 0, e_{X_2}(X_1, X_2) = 0\}$. When applying an action $\mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E_{add}}, \mathbf{E}_{del(e_{X_1})})$ on M where $\mathbf{E_{add}} = \{e'_{X_1}(X_1, X_2) = 0\}$, I have a non-recursive causal model with structural equations $\mathbf{E}_{X_1} = \mathbf{E} \cup \mathbf{E_{add}} \setminus \mathbf{E}_{del(e_{X_1})} = \{e'_{X_1}(X_1, X_2) = 0, e_{X_2}(X_1, X_2) = 0\}$. I define *recursive actions* with respect to a SEM $\mathbf{E}$ as follows.

Definition 5.1 (recursive action)

An action $A = \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ *on a SEM* $\mathbf{E}$ *is* recursive *if the manipulated model* $\mathbf{E}_A$ *is recursive and indeed represents the manipulated system.*

Notice that a recursive action is a self-contained action defined in Definition 4.5.

5.2.2 Recursive Actions on Systems with Irreversible Mechanisms

Given a causal model $M = \langle \mathbf{U}, \mathbf{V}, \mathbf{E} \rangle$ representing a system containing only irreversible mechanisms, a recursive atomic action $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E}_{add(X)}, \mathbf{E}_{del(e_X)})$, where $\mathbf{E}_{add(X)} = \{X = x\}$ and $e_X \in \mathbf{E}$, sets an endogenous variable $X \in \mathbf{V}$ to the value $x \in \mathsf{D}(X)$ and transforms M to the modified model $M_x = \langle \mathbf{U}, \mathbf{V}, \mathbf{E}_X \rangle$, where $\mathbf{E}_X = \mathbf{E} \cup \mathbf{E}_{add(X)} \setminus \mathbf{E}_{del(e_X)}$. Following Pearl's notation, I abbreviate $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E}_{add(X)}, \mathbf{E}_{del(e_X)})$ to $do(X = x)$. The effect of action $do(X = x)$ is given by M_x. Please note that the shorthand notation states that the operator $do(\cdot)$ always replaces e_X with its argument, a (probability) function that (probabilistically) assigns X with the value $x \in \mathsf{D}(X)$, to derive the modified model M_x and consequently its corresponding effects.

In addition to recursive atomic action, Pearl and Robins [1994; 1995] define three other types of recursive actions:

Conditional action $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E}_{del(e_X)})$ where $\mathbf{E_{pre}} = \{\mathbf{Z} = \mathbf{z} \mid \forall Z_j \in \mathbf{Z}, Z_j \notin \mathbf{Des}(X_i)\}$, $\mathbf{E_{add}} = \{X = x{\downarrow_{x=g(\mathbf{z})}}\}$. In Pearl's notation, a conditional action is denoted as $do(X = x{\downarrow_{x=g(\mathbf{z})}})$. The conditional action $do(X = x{\downarrow_{x=g(\mathbf{z})}})$ sets $X \in \mathbf{V}$ to the value $x = g(\mathbf{z})$ whenever $\mathbf{Z}$ attain values $\mathbf{z}$, where $g : \mathsf{D}(\mathbf{Z}) \rightarrow \mathsf{D}(X)$ and $\mathbf{Z}$ are non-descendants of X in $G(M)$.

Stochastic action $\mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E_{add}}, \mathbf{E}_{del(e_X)})$ where $\mathbf{E_{add}} = \{X = x{\downarrow_{\Pr^*(x)}}\}$. In Pearl's notation, a stochastic action is denoted as $do(X = x{\downarrow_{\Pr^*(x)}})$. The stochastic action $do(X = x{\downarrow_{\Pr^*(x)}})$ sets $X \in \mathbf{V}$ to the value x with probability $\Pr^*(x)$ where $\Pr^*(x)$ is specified externally.

Stochastic policy $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E}_{del(e_X)})$ where $\mathbf{E_{pre}} = \{\mathbf{Z} = \mathbf{z} \mid \forall Z_j \in \mathbf{Z}, Z_j \notin \mathbf{Des}(X_i)\}$ and $\mathbf{E_{add}} = \{X = x{\downarrow_{\Pr^*(x|\mathbf{z})}}\}$. In Pearl's notation, a stochastic policy is denoted as $do(X = x{\downarrow_{\Pr^*(x|\mathbf{z})}})$. The stochastic policy $do(X = x{\downarrow_{\Pr^*(x|\mathbf{z})}})$ sets $X = x$ with probability $\Pr^*(x|\mathbf{z})$ whenever $\mathbf{Z}$ attain values $\mathbf{z}$ where $\mathbf{Z}$ are non-descendants of X in $G(M)$ and $\Pr^*(x|\mathbf{z})$ is set externally.

Given a world $\langle M, u \rangle$, the *potential response* of $Y \in \mathbf{V}$ to action $do(\cdot)$ on variable $X \in \mathbf{V}$, denoted as $Y_{M_x}(\mathbf{u})$ or $Y_x(\mathbf{u})$, is the solution for Y to the set of equations $\mathbf{E}_x$ of M_x. $Y_x(\mathbf{u})$ can also be interpreted as the *counterfactual* value that Y would obtain had X been x in the counterfactual world brought about by action $do(\cdot)$. Given a probabilistic causal model $\langle M, \Pr(\mathbf{u}) \rangle$, the causal

effect on Y of an atomic action $do(X = x)$ is given by $\langle M_x, \Pr(\mathbf{u})\rangle$ as $\Pr(Y = y|do(X = x)) \equiv \Pr(y|\hat{x}) \equiv \Pr(Y_x = y) \triangleq \sum_{\{\mathbf{u}|Y_x(\mathbf{u})=y\}} \Pr(\mathbf{u})$. The causal effect on Y of a conditional action $do(X = x{\downarrow_{x=g(\mathbf{z})}})$ is expressed as $\Pr(Y = y|do(X = x{\downarrow_{x=g(\mathbf{z})}})) \equiv \Pr(y|\hat{x}){\downarrow_{x=g(\mathbf{z})}} \equiv \Pr(Y_x = y){\downarrow_{x=g(\mathbf{z})}} \triangleq \sum_{\mathbf{z}} \Pr(y|\hat{x}, \mathbf{z}){\downarrow_{x=g(\mathbf{z})}} \Pr(\mathbf{z})$. The causal effect on Y of a stochastic action $do(X = x{\downarrow_{\Pr^*(x)}})$ is expressed as $\Pr(Y = y|do(X = x{\downarrow_{\Pr^*(x)}})) \equiv \Pr(y|\hat{x}){\downarrow_{\Pr^*(x)}} \equiv \Pr(Y_x = y){\downarrow_{\Pr^*(x)}} \triangleq \sum_x \Pr(y|\hat{x}) \Pr^*(x)$. The causal effect on Y of a stochastic policy $do(X = x{\downarrow_{\Pr^*(x|\mathbf{z})}})$ is expressed as $\Pr(Y = y|do(X = x{\downarrow_{\Pr^*(x|\mathbf{z})}})) \equiv \Pr(y|\hat{x}){\downarrow_{\Pr^*(x|\mathbf{z})}} \equiv \Pr(Y_x = y){\downarrow_{\Pr^*(x|\mathbf{z})}} \triangleq \sum_x \sum_{\mathbf{z}} \Pr(y|\hat{x}, \mathbf{z}) \Pr^*(x|\mathbf{z}) \Pr(\mathbf{z})$.

Example 5.2 Suppose the model in Example 5.1 is an enemy's command center and one objective is to disrupt the enemy's communications. We can act on the communications C by, for example, jamming the signal with noise, and by setting this C to *damaged*. The modified causal model and its corresponding causal graph are shown in Figure 5.2. Please note that the intervention makes the arcs coming into C inactive (arc $G \rightarrow C$) □

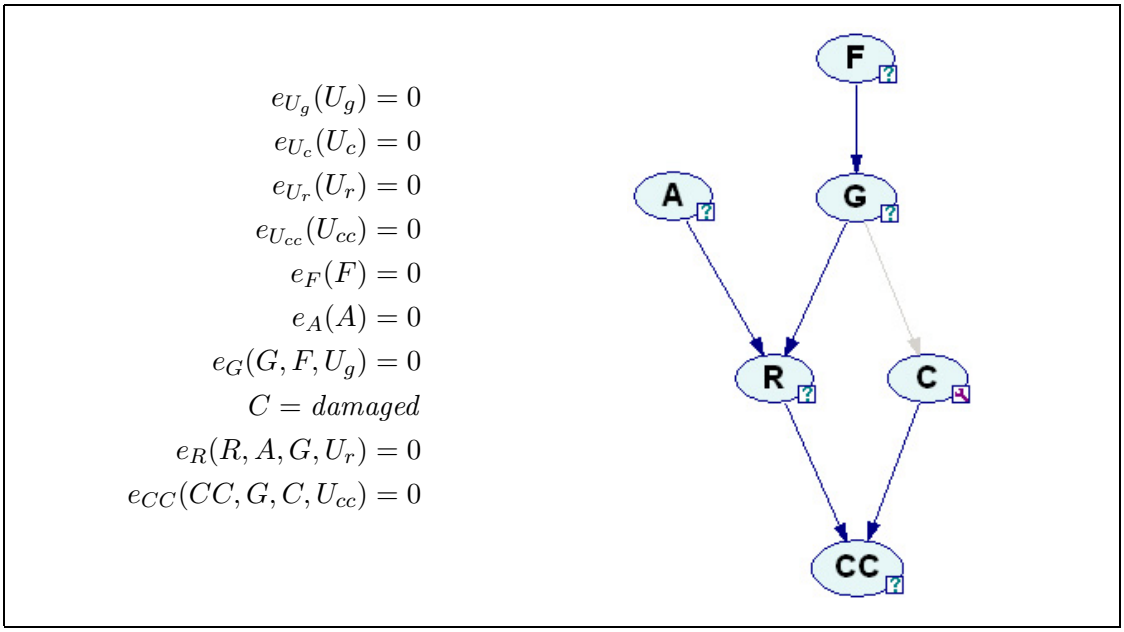


Figure 5.2. The modified causal model and its corresponding causal graph after an atomic action $do(C = damaged)$.

5.3 Augmented Models

In order to describe a decision problem at hand, I propose to augment a probabilistic causal model $\langle M, \Pr(\mathbf{u})\rangle$ by specifying properties over its variables. In addition to the *manipulability* and *observability* introduced in Section 3.3, I associate the property of *focus* (focus or non-focus) for each variable in a model. Let $\mathbf{X}$ denote all variables in M. A variable $X_i \in \mathbf{X}$ is *observable*, denoted as $X_i.o$, if it represents an entity that can be *measured* directly; *unobservable*, denoted as $X_i.\overline{o}$, otherwise. A variable $X_i \in \mathbf{X}$ is *manipulable*, denoted as $X_i.m$, if it represents an entity that can be *manipulated* directly; *non-manipulable*, denoted as $X_i.\overline{m}$, otherwise. I assume that a manipulable variable is always observable, i.e., I assume that I can always observe the effect of my manipulation. A variable $X_i \in \mathbf{X}$ is a *focus* variable, denoted as $X_i.f$, if it represents a decision objective; *non-focus*, denoted as $X_i.\overline{f}$, otherwise.

The goal of this work is to build a system that suggests decisions. At any stage of working with the system, there may be variables on which the user has decided to manipulate, whether based on the system's suggestions or the user's prior choices. I represent such pre-specified decisions by augmenting the model with a set of *decision* variables $\mathbf{D}$ along with their corresponding settings. The domain of each $D_i \in \mathbf{D}$ consists of the choices of setting $X_i.m$, the augmented manipulable variable, to a value $x_i \in \mathsf{D}(X_i)$, denoted as x_i', and a special state *idle* representing the force of nature [Pearl, 1993]. Let $\mathbf{Pa}(X_i)$ denote the set of parents of X_i in $G(M)$, i.e., $\mathbf{Pa}(X_i) = \mathbf{Vars}(e_{X_i}) \setminus \{X_i\}$. I augment the equation $e_{X_i}(X_i, \mathbf{Pa}(X_i)) = 0$ to $e'_{X_i}(X_i, \mathbf{Pa}'(X_i)) = 0$, where $\mathbf{Pa}'(X_i) = \mathbf{Pa}(X_i) \cup \mathbf{Z} \cup \{D_i\}$ and $\mathbf{Z} \subset \mathbf{X}$, a set of non-descendants of X_i in $G(M)$ brought about by interventions. I define the augmented equation $e'_{X_i}(X_i, \mathbf{Pa}'(X_i)) = 0$ as

$$e'_{X_i}(X_i, \mathbf{Pa}'(X_i)) \triangleq \quad \begin{cases} e^*_{X_i}(X_i, \mathbf{Pa}'(X_i)) = 0 & \text{if } D_i = x_i', \\ e_{X_i}(X_i, \mathbf{Pa}(X_i)) = 0 & \text{if } D_i = \mathit{idle}, \end{cases} \tag{5.1}$$

where the form of $e^*_{X_i}(X_i, \mathbf{Pa}'(X_i)) = 0$ depends on the type of intervention (See Table 5.1.). To represent *concurrent actions* on X_i and X_j, in addition to D_i and D_j and corresponding augmentations on e_{X_i} and e_{X_j}, I add a decision variable, denoted as D_{ij}, to represent the concurrency. The domain of D_{ij} is $\mathsf{D}(D_{ij}) = \mathsf{D}(D_i) \times \mathsf{D}(D_j)$. I add *projection* equations $e_{D_i} : \mathsf{D}(D_{ij}) \to \mathsf{D}(D_i)$

and $e_{D_j} : \mathsf{D}(D_{ij}) \rightarrow \mathsf{D}(D_j)$ such that $D_i = d_i$ and $D_j = d_j$ for each $d_{ij} \in \mathsf{D}(D_{ij})$.

Table 5.1. The form of $e^*_{X_i}(X_i, \mathbf{Pa}'(X_i))$ characterized with respect to different types of interventions.

	$e^*_{X_i}(X_i, \mathbf{Pa}'(X_i)) = 0$
Atomic	$\Pr(x_i \mid \mathbf{pa}'(X_i)) = \begin{cases} 1 & \text{if } x'_i = x_i, \\ 0 & \text{otherwise.} \end{cases}$
Conditional	$\Pr(x_i \mid \mathbf{pa}'(X_i)) = \begin{cases} 1 & \text{if } x'_i = x_i \text{and } x_i = g(\mathbf{z}), \\ 0 & \text{otherwise.} \end{cases}$
Stochastic	$\Pr(x_i \mid \mathbf{pa}'(X_i)) = \begin{cases} \Pr^*(x_i) & \text{if } x'_i = x_i, \\ 0 & \text{otherwise.} \end{cases}$
Policy	$\Pr(x_i \mid \mathbf{pa}'(X_i)) = \begin{cases} \Pr^*(x_i \mid \mathbf{z}) & \text{if } x'_i = x_i \text{and } x_i = g(\mathbf{z}), \\ 0 & \text{otherwise.} \end{cases}$

Finally, to represent the preferences over the given set of objectives and decisions, I augment the model by a set of *utility* variables $\mathbf{UT}$ along with their utility functions $\mathbf{U}$. Each utility function $\mathsf{U}_i \in \mathbf{U}$ can only have focus or decision variables as its arguments. Formally, I can define an augmented model as follows.

Definition 5.2 (Augmented Model)

An augmented model for a decision problem is $M_A = \langle \langle M, \Pr(\mathbf{u}) \rangle, C(\mathbf{X}), \langle \mathbf{D}, \mathbf{E}' \rangle, \langle \mathbf{UT}, \mathbf{U} \rangle \rangle$*, where:*

1. $\langle M, \Pr(\mathbf{u}) \rangle$ *is a probabilistic causal model.*
2. $C(\mathbf{X})$ *is a characterization of observability, manipulability, and focus for each* $X_i \in \mathbf{X}$*.*
3. $\langle \mathbf{D}, \mathbf{E}' \rangle$ *is a set of decision variables* $\mathbf{D}$ *and the modified equation* $\mathbf{E}'$ *with respect to the decisions.*
4. $\langle \mathbf{UT}, \mathbf{U} \rangle$ *is a set of utility variables* $\mathbf{UT}$ *and its corresponding utility functions* $\mathbf{U}$ *over a set of focus variables, characterized by* $C(\mathbf{X})$*, and a subset of decision variables in* $\mathbf{D}$*.*

Example 5.3 Suppose variables F, A, G, R, C, and CC are manipulable and CC is the only focus variable in Example 5.1. We add a utility node, *Utility*, with utility function $\mathsf{U}(CC)$, to represent our preference over the states of CC. The corresponding causal graph is shown in Figure 5.3 (a). Suppose we have a decision option of manipulating the communications C with no direct influence on *Utility*. We then have a causal graph with D_c as a decision variable shown in Figure 5.3 (b). □

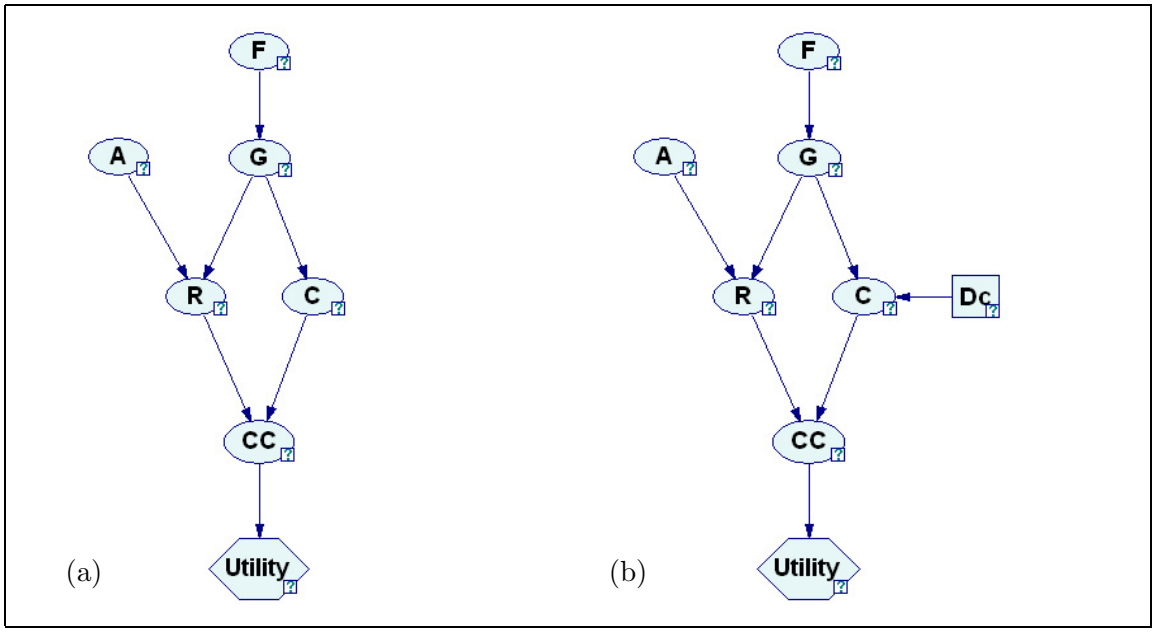


Figure 5.3. (a) Example model augmented with utility over variable CC. (b) The model in (a) augmented with the atomic action D_c.

I emphasize that specifying that a variable is manipulable, which merely acknowledges the possibility of interventions, is not the same as designating a decision variable in an influence diagram, which requires the explicit specifications of a decision variable along with its consequences. Declaring that a variable is manipulable allows the algorithm that searches for opportunities to explore possible interventions that might not have been foreseen when the model was constructed. Furthermore, I require neither a manipulable variable being intervened upon, nor an observable variable being observed. It is the task of search for opportunities and information gathering to determine which variable one should intervene on or observe and in what order. In other words, I propose to relax not only the assumption of a fixed sequence of intervening actions and observations in influence diagrams, but also the assumption of a fixed operation over a variable. For example, a manipulable variable may be intervened on, observed, or unknown, depending on different decision sequences generated by search for opportunities and information gathering. Only when a manipulable variable is augmented by the decision variable and its augmented equation as in the form of Equation 5.1, one commits to intervene on the manipulable variable with one of the policies specified by the decision variable.

5.4 Value of Intervention

Suppose that I am considering an additional atomic intervention on an unaugmented manipulable variable X_k in an augmented model M_A. I augment M_A by adding a new decision variable D_k and modify e_{X_k} to e'_{X_k}, as demonstrated in Equation 5.1. Let $\mathbf{D'} = \mathbf{D} \cup \{D_k\}$ be the new set of decision variables. I also augment utility function $\mathbf{U}$ to $\mathbf{U'}$ if the intervention directly influences $\mathbf{U}$. Most interventions come at a certain cost and the cost of intervention can be incorporated by augmenting $\mathbf{U}$. Let M'_A denote the newly augmented model, $\pi(M'_A)$ its optimal strategy, and $\mathsf{MEU}(M_A)$ and $\mathsf{MEU}(M'_A)$ the maximum expected utility yielded by the optimal strategies $\pi(M_A)$ and $\pi(M'_A)$ respectively. I define the *value of intervention* on X_k as

$$\mathsf{VOINT}(D_k = d_k^*) \triangleq \mathsf{MEU}(M'_A) - \mathsf{MEU}(M_A), \tag{5.2}$$

where $d_k^* \in \mathsf{D}(D_k)$ is yielded in $\pi(M'_A)$ by the optimal policy of D_k. Note that value of atomic intervention can account for the concept of the value of control in influence diagrams. Since augmented models support prediction of the effect of actions, I am not constrained to only atomic interventions (control) on nodes with no predecessors as in the case of value of control in influence diagrams. To compute value of intervention for a conditional, stochastic, or stochastic policy action, I simply substitute the action of interest for the atomic intervention in augmenting model M_A and perform analysis using Equation (5.2).

Theorem 5.1 *Let M_A be an augmented model and X_k be an unaugmented manipulable variable in M_A. If M'_A is the augmented model of M_A resulting from considering an intervention on X_k that has no direct impact on utility function U, then $\mathsf{MEU}(M'_A) \geq \mathsf{MEU}(M_A)$.*

Proof: When evaluating $\pi(M_A)$, I can decompose the joint probability distribution of M_A according to $G(M_A)$. When considering an additional intervention on X_k that has no direct impact on U, I augment M_A into M'_A by modifying $\Pr(x_k|\mathbf{pa}(X_k))$ to $\Pr(x_k|\mathbf{pa}'(X_k))$. Now, when evaluating $\pi(M'_A)$, I also decompose the joint probability distribution of M'_A according to the $G(M'_A)$. Notice

that X_k participates in $\mathsf{MEU}(M_A)$ as

$$\mathsf{MEU}(M_A) = \cdots \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k|\mathbf{pa}(X_k)) \cdots \mathsf{U},$$

and in $\mathsf{MEU}(M'_A)$ as

$$\mathsf{MEU}(M'_A) = \cdots \max_{d_k \in \mathsf{D}(D_k)} \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k|\mathbf{pa}'(X_k)) \cdots \mathsf{U},$$

where $d_k \in \mathbf{pa}'(X_k)$. Since $\Pr(x_k|\mathbf{pa}(X_k))$ is also represented in $\Pr(x_k|\mathbf{pa}'(X_k))$ as $d_k = idle$, $\mathsf{MEU}(M'_A) \geq \mathsf{MEU}(M_A)$ according to the maximization operator. □

Consider a simple model M_A which consists of one variable X_k with probability distribution $\Pr(x_k)$, where $x_k \in \mathsf{D}(X_k)$, and a utility variable UT with utility function $\mathsf{U}(x_k)$. We have $\mathsf{MEU}(M_A) = \mathsf{EU}(M_A) = \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k)\mathsf{U}(x_k)$. Consider an additional stochastic intervention on X_k that has no direct impact on U. We augment M_A to M'_A with a new decision variable D_k with domain $\mathsf{D}(D_k) = \mathsf{D}(X_k) \cup \{idle\}$, and modify the probability distribution of X_k to

$$\Pr(x_k|d_k) = \begin{cases} \Pr^*(x_k) & \text{if } D_k = x'_k \text{ and } x_k = x'_k, \\ 0 & \text{if } D_k = x'_k \text{ and } x_k \neq x'_k, \\ \Pr(x_k) & \text{if } D_i = idle. \end{cases}$$

We have

$$\mathsf{MEU}(M'_A) = \max_{d_k \in \mathsf{D}(D_k)} \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k|d_k)\mathsf{U}(x_k),$$

and the optimal value of setting D_k,

$$d_k^* = \arg\max_{d_k \in \mathsf{D}(D_k)} \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k|d_k)\mathsf{U}(x_k).$$

It shows that d_k^* is taken on one of x'_k only if

$$\sum_{x_k \in \mathsf{D}(X_k)} \Pr^*(x_k)\mathsf{U}(x_k) \geq \sum_{x_k \in \mathsf{D}(X_k)} \Pr(x_k)\mathsf{U}(x_k).$$

In other words, it suggests not to act if the stochastic intervention under consideration does no better than the nature. By the same token, d_k^* can always take on the state *idle* for other types of interventions that have no direct impacts on utility functions U, if it will not do better than the nature. Next, consider an augmented model with variables $D_A \rightarrow A \rightarrow B \rightarrow U$. We have

$$\mathsf{MEU}(M_A) = \max_{d_A \in \mathsf{D}(D_A)} \sum_{a \in \mathsf{D}(A)} \Pr(a|d_A) \sum_{b \in \mathsf{D}(B)} \Pr(b|a)\mathsf{U}(b).$$

Consider an additional intervention on B with no direct impacts on U and augment M_A to M'_A correspondingly. We have

$$\mathsf{MEU}(M'_A) = \max_{d_A \in \mathsf{D}(D_A)} \sum_{a \in \mathsf{D}(A)} \Pr(a|d_A) \max_{d_B \in \mathsf{D}(D_B)} \sum_{b \in \mathsf{D}(B)} \Pr(b|a, d_B)\mathsf{U}(b).$$

We see, again, that D_B will take on the state other than *idle* only if

$$\sum_{b \in \mathsf{D}(B)} \Pr(b|a, d_B)\mathsf{U}(b) > \sum_{b \in \mathsf{D}(B)} \Pr(b|a)\mathsf{U}(b).$$

Example 5.4 Suppose that we are considering an additional atomic intervention on R in the model of Figure 5.3 (b). Suppose that the atomic intervention under consideration is associated with a cost of intervention function $\mathsf{CI}(Dr)$ over $\mathsf{D}(D_R)$. We add a multi-attribute utility function MAU that combines $\mathsf{CI}(Dr)$ and *Utility*. The augmented model is shown in Figure 5.4 (a). The value of atomic intervention $\mathsf{VOINT}(D_R = d_R^*) = \mathsf{MEU}(M'_A) - \mathsf{MEU}(M_A)$, where M'_A is the model in Figure 5.4 (a) and M_A is the one in Figure 5.3 (b). □

5.5 Search for Opportunities

Search for opportunities refers to the problem of identifying novel interventions that can improve the outcomes. Ideally, one should consider all possible novel interventions on all unaugmented manipulable variables simultaneously, along with existing decisions, to find the optimal strategy and the maximum expected utility for the model. In a complex system, however, such analysis can easily challenge the modeling and computational complexity. For example, even if we constrain ourselves

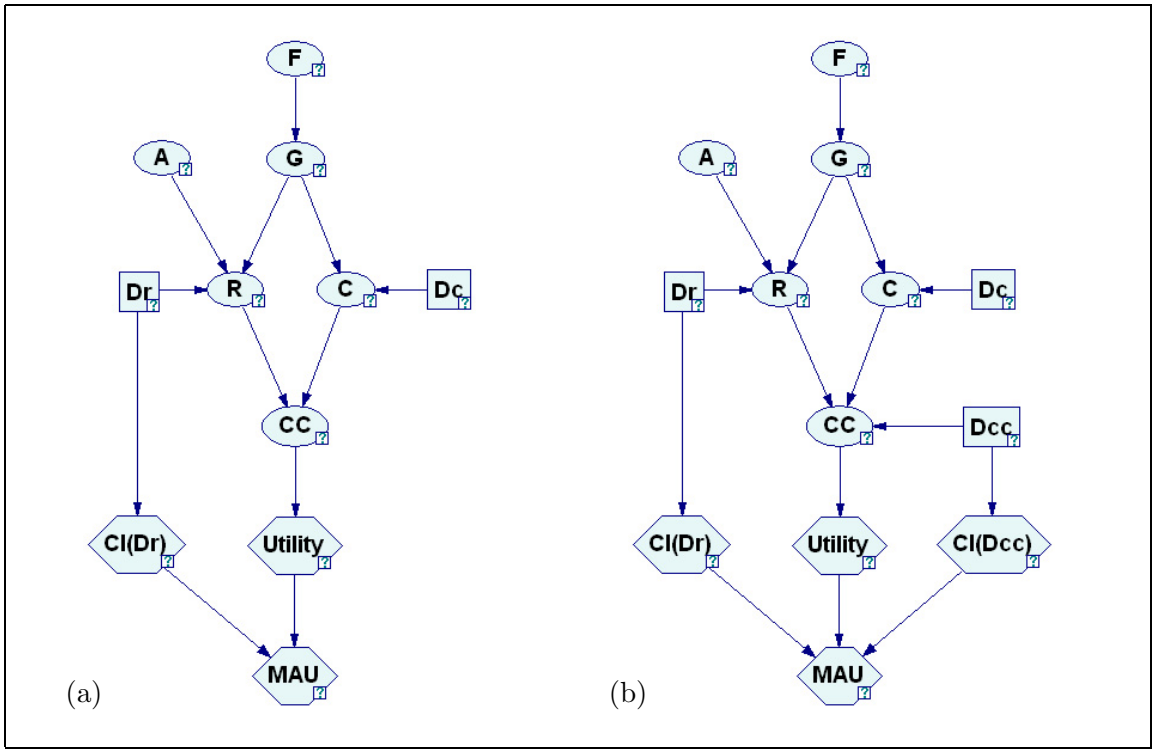


Figure 5.4. (a) The model augmented with a possible atomic intervention on R, the cost of intervention $\mathsf{CI}(D_R)$, and the multi-attribute utility function MAU. (b) The optimal atomic intervention on R is instantiated by setting D_R to d_R^* and the model is augmented with a possible intervention on CC and the cost of intervention $\mathsf{CI}(D_{CC})$ in the myopic approach of search for opportunities.

to considering only recursive atomic interventions on manipulable variables in the command center example, theoretically, we need to elicit utilities and to evaluate strategies for 3^6 combinations of all possible atomic interventions. In general, if we have n manipulable variables with m states for each, we will have $(m+1)^n$ combinations of utilities and strategies, including one extra dimension for the force of nature.

To simplify the problem of modeling, I assume that all novel interventions under consideration have no direct impact on utility functions except by the cost of intervention. One can elicit the cost of intervention $\mathsf{CI}(D_k)$ for each novel intervention on X_k. Let $\mathbf{CI}$ denote all costs of interventions under consideration. I assume that my multi-attribute utility function over the existing individual utility functions $\mathbf{U}$ and costs of interventions is *decomposable*, i.e., there exists a multi-attribute utility function MAU that takes as arguments each $\mathsf{U}_i \in \mathbf{U}$ and each $\mathsf{CI}(D_k) \in \mathbf{CI}$ and combine them

Procedure *MyopicSFOIntSysIrrevMechs*$(M_A, \delta^*, \mathbf{X}^m, \mathsf{CI})$

Input: An augmented model M_A, a threshold δ^* for the increase of expected utility, and the set of unaugmented manipulable variables $\mathbf{X}^m$ and its cost function $\mathsf{CI}(D_i)$ for each $X_i \in \mathbf{X}^m$ in M_A.

Output: A sequence of interventions $\mathbf{D}$ on the subset of $\mathbf{X}^m$.

1. $\mathbf{D} := \emptyset$; $M^* := M_A$; *Update*(M^*); $\mu^* := \mathsf{MEU}(M^*)$; *found* := **false**;
2. **while** *found* = **false**
3. $M := M^*$; $\mu := \mu^*$; $\delta := \delta^*$;
4. **for each** $X_i \in \mathbf{X}^m$
5. $M' :=$ *Augment*$(M^*, do(X_i), \mathsf{CI}(D_i))$; *Update*$(M')$; $\mu' := \mathsf{MEU}(M')$;
6. $\triangle\mu := \mu' - \mu^*$; /* $\mathsf{VOINT}(D_i)$ */
7. **if** $\triangle\mu > \delta$ **then**
8. $\delta := \triangle\mu$; $M := M'$; $\mu := \mu'$; $X := X_i$; $d := \pi(M_i')$;
9. *found* := **true**;
10. **end if**
11. **end for each**
12. **if** *found* = **true then**
13. $M^* :=$ *Instantiate*(M, X, d); $\mu^* := \mu$; $\mathbf{D} := \mathbf{D} \cup \{(X, d)\}$;
14. *found* := **false**;
15. **else** *found* := **true**;
16. **end if**
17. **end while**
18. **return D**;

Figure 5.5. Myopic search for opportunities with intervening actions on systems containing only irreversible mechanisms. *Update*(M) computes the optimal strategy and maximum expected utility for a model M. *Augment*$(M, do(X_i), \mathsf{CI}(D_i))$ denotes the operation of augmenting the model M with an intervention on X_i with the cost of intervention $\mathsf{CI}(D_i)$. *Instantiate*(M, X, d) denotes the operation of setting the value of D_X to d in M.

with a functional form (such as a simple linear or a multiplicative form). In the case of a linearly additive MAU, we need to elicit $n \times (m + 1)$ numbers for CI and at most $n + 1$ numbers for the

weights in MAU (if the units of cost are the same, this number can be significantly smaller). Next, I approximate the optimal strategy computation by the *myopic* (greedy) search that considers one intervention at a time and selects the one with the maximum value of intervention to perform. I act according to the selected intervention and perform the myopic search again to select the next intervention to act until there is no intervention that can improve the maximum expected utility over the predefined threshold. Figure 5.5 outlines this procedure. The complexity of this procedure is NP-hard since it relies on the probabilistic belief update in Line 1 and 5.

Example 5.5 Given the model in Figure 5.3 (b), suppose that we use the myopic approach to identify that R is the next variable to intervene by an atomic intervention. The model is augmented with an intervention on R as shown in Figure 5.4 (a). After intervening on R by setting D_R to d_R^*, suppose that we identify that CC is the next variable to act on by an atomic intervention. We have the model augmented as shown in Figure 5.4 (b). □

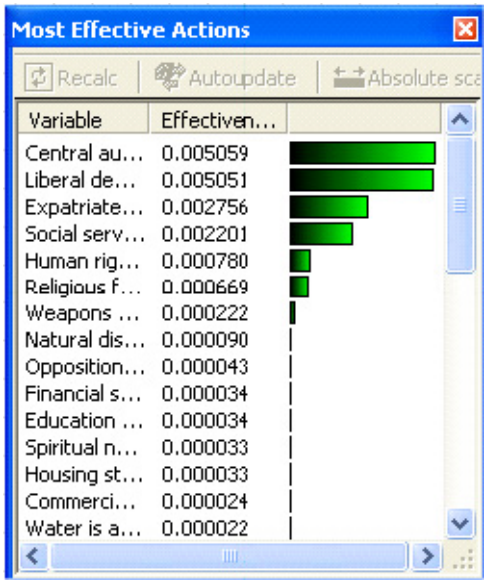


Figure 5.6. A ranked list of value of interventions on unaugmented manipulable variables.

The procedure in Fig. 5.5 can be applied by a robot to find out the next most effective action. It can also be a useful extension of a modeling environment, which is how I plan to apply it. As illustrated in Figure 5.6, I present users with a list of ranked values of interventions, generated by Lines 4-11 in Figure 5.5.[1] Users may take the suggestion from the myopic search to perform

[1] As far as utility and cost of intervention are concerned, I use a simple linearly additive form of MAU function.

the intervention at the top of the list, or select any other intervention from the list to alter the generation of the decision sequences. Once the users have entered the intervention into the system, the system performs the myopic search again to update the ranked list of possible interventions. This interactive environment allows the users also to perform "what if" analysis in generating decision sequences.

5.6 Non-intervening Action and Value of Observation

In many decision problems, we bring new mechanisms into the model when we consider to apply a *non-intervening* action on a system. For example, consider a model that describes the relations among heart disease (HD), blood pressure (BP), and headache (HA) as mechanisms $f_{HD}(HD) = 0$, $f_{BP}(HD, BP) = 0$, and $f_{HA}(BP, HA) = 0$. Assume that the utility (*Utility*) of a patient directly depends on headache (HA). The causal graph for this example is depicted in Figure 5.7(a). An example of a non-intervening action would be measuring blood pressure (MBP), which brings the variable blood pressure reading (BPR) and the mechanism describing how the blood pressure is measured, $f_{BPR}(BP, BPR, MBP) = 0$, into consideration. I represent the cost of measuring blood pressure ($\mathsf{CO}(MBP)$) as a value function of MBP, i.e., $\mathsf{U}(MBP)$. Now, I have the causal graph as depicted in Figure 5.7(b).

Let M'_A denote the augmented model of M_A when considering a non-intervening action on an observable variable in M_A. Let $\pi(M'_A)$ and $\pi(M_A)$ be the optimal strategy of M'_A and M_A respectively. I can define the *value of observation* (value of information) as

$$\mathsf{VOOBS}(D_k = d_k^*) \triangleq \mathsf{MEU}(M'_A) - \mathsf{MEU}(M_A),$$

where $d_k^* \in \mathsf{D}(D_k)$ is yielded in $\mathsf{MEU}(M'_A)$ by the optimal policy of D_k.

I can now choose between intervening actions and non-intervention actions by comparing their values of interventions and observations. For example, I can compute the value of observation for non-intervening actions such as measuring blood pressure (MBP) based on model depicted in Figure 5.7(a) and (b). I can then compute the value of interventions for intervening actions such as taking the medicine for blood pressure (D_{bp}), taking the medicine for headache (D_{ha}), or taking

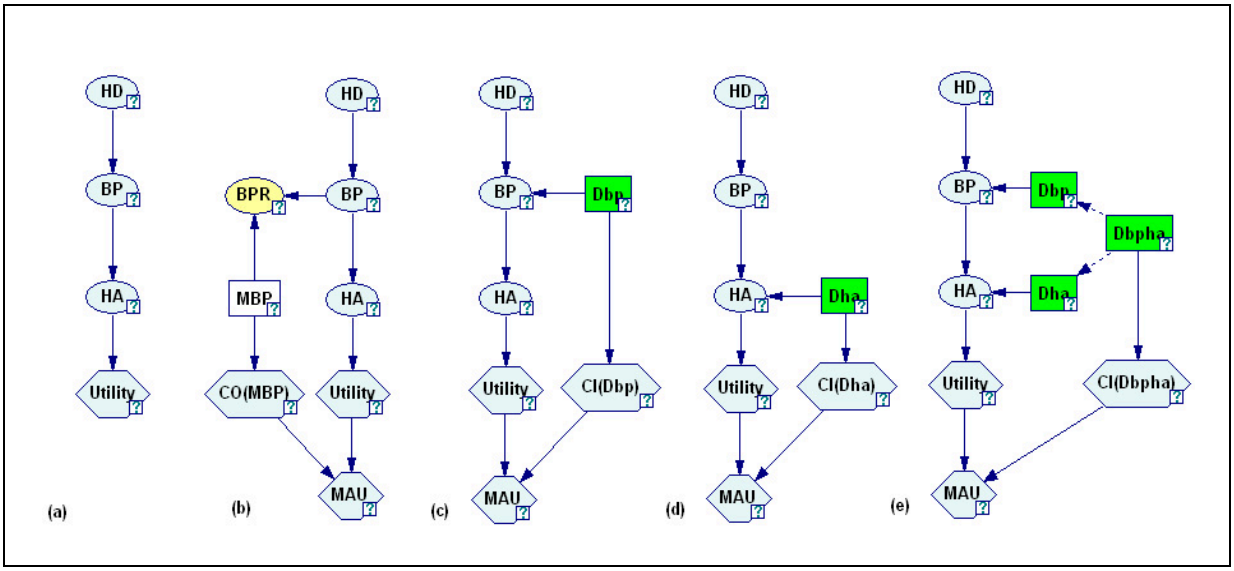


Figure 5.7. (a) depicts the causal relations among heart disease (HD), blood pressure (BP), and headache (HA); (b) depicts the augmented model for a non-intervening action – measuring blood pressure (MBR), its reading (BPR), and cost ($\mathsf{CO}(MBR)$); (c) depicts the augmented model for the intervening action – taking the blood pressure control medicine (D_{bp}) and its cost ($\mathsf{CI}(D_{bp})$); (d) depicts the augmented model for intervening action – taking the medicine for headache (D_{ha}) and its cost ($\mathsf{CI}(D_{ha})$); (e) depicts the augmented model for concurrent intervening action – taking the medicine for both blood pressure and headache (D_{bpha}) and its cost ($\mathsf{CI}(D_{bpha})$).

medicine for both blood pressure and headache (D_{bpha}), based on model depicted in Figure 5.7(c), (d), (e), and (a) respectively.

5.7 Search for Opportunities in Systems Containing Only Irreversible Mechanisms

Very often we act because of what we have observed, since we intend to change the world in the desired direction. We therefore have a sequence of non-intervening and intervening actions. In this section, I address the problem of search for opportunities in system containing only irreversible mechanisms.

5.7.1 Persistence and Response

I now continue the example in Section 5.6. After examining the reading of a patient's blood pressure, a doctor may prescribe a medicine to control the blood pressure such that the symptom of

headache can be eased. Consequently, I need to augment the model to represent the *persistency* of heart disease, which has not been treated, and the *response* of blood pressure and headache relative to the prescribed blood pressure control medicine. It is common sense that the previous reading of blood pressure becomes invalid after taking the blood pressure control medicine. However, the reading of blood pressure before taking the medicine should affect our belief of the severity of the heart disease and its persistence.

In Figure 5.8(a), I depict a model representing the relations among heart disease (HD), blood pressure (BP), and headache (HA). Since a non-intervening action does not intervene into mechanisms in a model, I augment the model in 5.8(a) with the non-intervening action, measuring blood pressure (MBP), and the mechanisms brought about by the non-intervening action, the blood pressure reading (BPR) and the observation cost $\mathsf{CO}(MBP)$. Figure 5.8(b) shows the augmented model. To model the intervening action of taking a blood pressure control medicine (D_{bp}) conditioned on the non-intervening action (MBP) and its reading (BPR), I copy the mechanisms describing the relations among heart disease, blood pressure, and headache ($HD \rightarrow BP \rightarrow HA$) to the next time slice as ($HD' \rightarrow BP' \rightarrow HA'$) (See Figure 5.8(c)). Then I augment the definition of heart disease in the next time slice (HD') to indicate that it is determined by the heart disease in the previous time slice (HD). In other words, the arc in $HD \rightarrow HD'$ represents the persistency of the heart disease. Then I augment the model by adding the conditional intervening action – taking the blood pressure control medicine (D_{bp}), which is conditioned on the blood pressure reading (BPR) and the non-intervening action (MBP), and by modifying the distribution of (BP') to depends on HD' and D_{bp}. Please note that the instantiated observation of blood pressure reading (BPR) changes the belief of heart disease (HD) and then changes the belief or (HD') through the persistent relation between (HD) and (HD'). In other words, the evidence of blood pressure reading (BPR) is not only used for making the decision of blood pressure control (D_{bp}) but also used in updating the belief between time slices through the persistence link between $HD \rightarrow HD'$. Furthermore, I drop the headache and its utility in the previous time slice since we are now interested in the utility in the next time slice.

To see how the model structures are changed with respect to the sequential non-intervening and intervening actions, I first show the set of structural equations $\mathbf{E}$ for the model M on the left. After applying the non-intervening action $A_{MBP} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ where $\mathbf{E_{pre}} = \{BP \in$

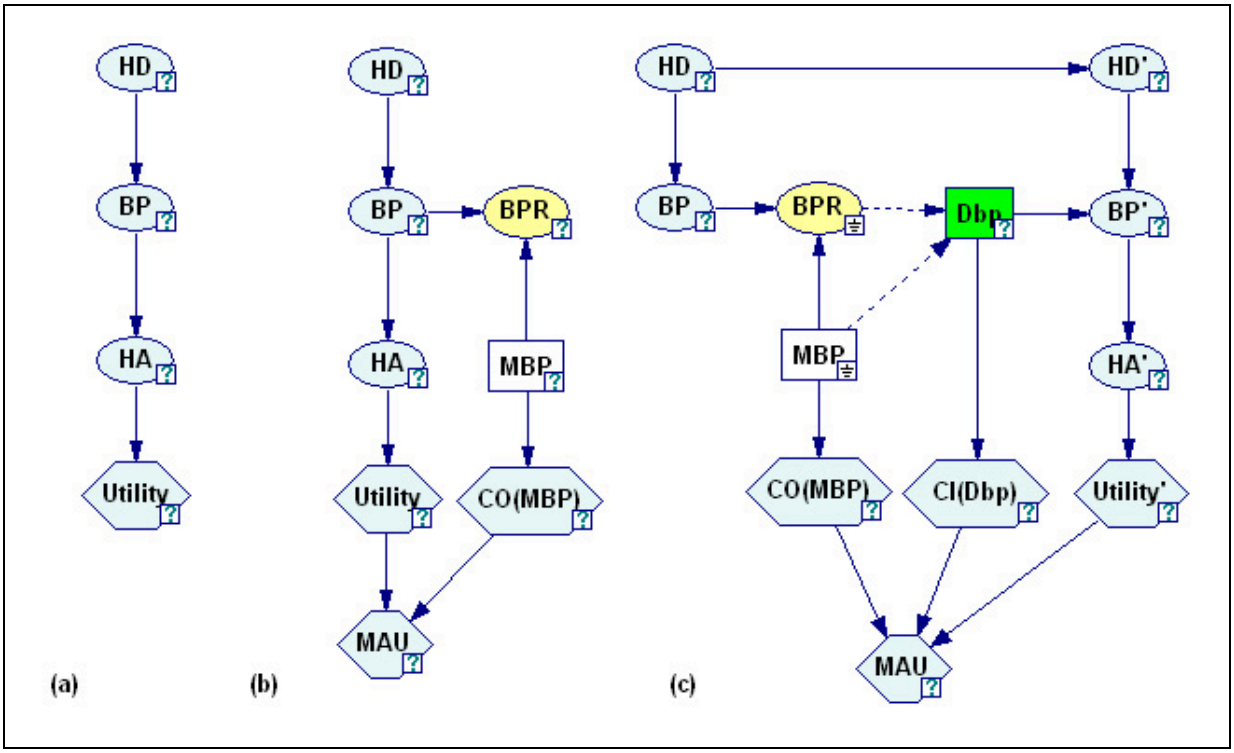


Figure 5.8. (a) depicts relations among heart disease (HD), blood pressure (BP), and headache (HA); (b) depicts the augmented model for the non-intervening action – measuring blood pressure (MBR), its reading (BPR) and cost ($\mathsf{CO}(MBR)$); (c) depicts the augmented model for considering taking the blood pressure control medicine (D_{bp}) and its cost ($\mathsf{Cl}(D_{bp})$), after measuring the blood pressure.

$\mathbf{Vars}(\mathbf{E})\}$, $\mathbf{E_{add}} = \{f_{BPR}, f_{\mathsf{CO}(MBP)}\}$, and $\mathbf{E_{del}} = \emptyset$, the set of modified structural equations is shown at the right hand side.

$$\begin{cases} f_{HD}(HD) = 0 \\ f_{BP}(BP, HD) = 0 \\ f_{HA}(HA, BP) = 0 \\ f_{Utility}(Utility, HA) = 0 \end{cases} \Longrightarrow \begin{cases} f_{HD}(HD) = 0 \\ f_{BP}(BP, HD) = 0 \\ f_{HA}(HA, BP) = 0 \\ f_{Utility}(Utility, HA) = 0 \\ f_{BPR}(BPR, MBP, BP) = 0 \\ f_{MBP}(MBP) = 0 \\ f_{\mathsf{CO}(MBP)}(\mathsf{CO}(MBP), MBP) = 0 \end{cases}$$

We see that A_{MBP} brought f_{BPR} and $f_{\mathsf{CO}(MBP)}$ into the model but did not intervene into any of the existing mechanisms: f_{HD}, f_{BP}, f_{HA}, and $f_{Utility}$. Furthermore, A_{MBP} is applicable only

when the variable is about to be observed is in the model, i.e., $\mathbf{E_{pre}} = \{BP \in \mathbf{Vars}(\mathbf{E})\}$.

Let $M_{A_{MBP}}$ denote the augmented model after applying A_{MBP} on M and $\mathbf{E}_{A_{MBP}}$ denote the set of structural equations in $M_{A_{MBP}}$. Consider the intervening action $A_{D_{bp}} \triangleq \mathbf{Act}(\mathbf{E}_{\mathbf{A}_{MBP}}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ where $\mathbf{E_{pre}} = \{MBP \in \mathbf{Vars}(\mathbf{E}) \wedge MBP = true, BPR \in \mathbf{Vars}(\mathbf{E}) \wedge BPR = high, BP \in \mathbf{Vars}(\mathbf{E})\}$, $\mathbf{E_{add}} = \{f_{D_{bp}}, f_{HD'}, f_{BP'}, f_{HA'}, f_{Utility'}, f_{\mathsf{CI}(D_{bp})}\}$, and $\mathbf{E_{del}} = \{f_{HA}, f_{Utility}\}$. The set of modified structural equations is shown at the right hand side.

$$\begin{cases} f_{HD}(HD) = 0 \\ f_{BP}(BP, HD) = 0 \\ f_{HA}(HA, BP) = 0 \\ f_{Utility}(Utility, HA) = 0 \\ f_{BPR}(BPR, MBP, BP) = 0 \\ f_{MBP}(MBP) = 0 \\ f_{\mathsf{CO}(MBP)}(\mathsf{CO}(MBP), MBP) = 0 \end{cases} \Longrightarrow \begin{cases} f_{HD}(HD) = 0 \\ f_{BP}(BP, HD) = 0 \\ f_{BPR}(BPR = high, MBP = true, BP) = 0 \\ f_{MBP}(MBP) = 0 \\ f_{\mathsf{CO}(MBP)}(\mathsf{CO}(MBP), MBP) = 0 \\ f_{D_{bp}}(D_{bp}, BPR = high, MBP = true) = 0 \\ f_{HD'}(HD', HD) = 0 \\ f_{BP'}(BP', HD', D_{bp}) = 0 \\ f_{HA'}(HA', BP') = 0 \\ f_{Utility'}(Utility', HA') = 0 \\ f_{\mathsf{CI}(D_{bp})}(\mathsf{CI}(D_{bp}), D_{bp}) = 0 \end{cases}$$

As we can see, the persistency of heart disease between time slice is modeled by the structural equation $f_{HD'}$. The previous reading of blood pressure ($BPR = high$) updates our belief of the severity of heart disease in the next time slice (HD') through the path of persistence ($HD \rightarrow HD'$). The decision of taking blood pressure control medicine (D_{bp}) depends on the decision of measuring the blood pressure ($MBP = true$) and the reading of blood pressure ($BPR = high$). The blood pressure after the intervention is governed by the structural equation $f_{BP'}$. The headache at the next time slice (HA') is governed by the structural equation $f_{HA'}$.

5.7.2 Generic Actions

After presenting the way of augmenting a model for a non-intervening action followed by an intervening action, I examine the generality of both A_{MBP} and $A_{D_{bp}}$. The non-intervening action

A_{MBP} is applicable to all models in the same domain as long as the precondition $\mathbf{E_{pre}} = \{BP \in \mathbf{Vars}(\mathbf{E})\}$ is held, since it simply brings in new mechanisms into a model without intervening on the rest. On the other hand, we can see that $A_{D_{bp}}$ is not directly applicable to all models in the domain, since I have represented the *persistence* and *response* into $\mathbf{E_{add}}$ and $\mathbf{E_{del}}$ of $A_{D_{bp}}$. To make the intervening action $A_{D_{bp}}$ model independent, I shall specify the relations that are locally relevant to $A_{D_{bp}}$, namely $f_{D_{bp}}$, $f_{BP'}$, and $f_{\mathsf{Cl}(D_{bp})}$ in $\mathbf{E_{add}}$. The relation representing persistence, $f_{HD'}$ in $\mathbf{E_{add}}$, should be inferred from $\mathbf{E}_{A_{MBP}}$. Similarly, the relations representing response, $f_{HA'}$ and $f_{Utility'}$ in $\mathbf{E_{add}}$ and the $\mathbf{E_{del}}$ of $A_{D_{bp}}$, should be inferred from $\mathbf{E}_{A_{MBP}}$.

Before I give the framework for inferring persistence and response, I first discuss when I shall model a system into two consequent time slices. When I apply only intervening actions on systems containing only irreversible mechanisms, as in Section 5.5, there is no need to model a system in two consequent time slices, since no variables are observed and all intervening actions are applicable when their preconditions are satisfied. However, I need to model a system in two consequent time slices when I apply an intervening action on a direct or an indirect cause of an observed variable, by which I determine the intervening action. In other words, if I do not model a system in two consequent time slices, I cannot distinguish the variable which is a cause of observed variable from the same variable which is manipulated by the intervening action.

Second, I need to decide on which variables I should apply their persistence relations when I model a system in two consequent time slices. Since all endogenous variables are determined within the model, I shall only consider applying persistence relations for exogenous variables at the next time slice. In addition, I can elicit persistent relations before I decide what type of actions to apply on the system, since the persistence relations simply represent the evolutional influences from the system in the current time slice to the system in the next time slice. Furthermore, the persistence relations serve as the path way of carrying the information brought about by observations from the current time slice to the next time slice.

Third, I need to decide the responses resulting from an intervening action. Modeling a system in two different time slices is one of responses of an intervening actions. For such response, I copy mechanisms from the system in the current time slice into the system in the next time slice, but leave those invalid observations. The other response is to remove mechanisms, which are not needed for the decision on the system in the next time slice, from the system in the current time

slice. Such mechanisms are those govern the nodes which are not ancestors of the evidences and are d-separated from the evidences given the manipulated variable.

Procedure *InstantiateActionIrrev*$(\mathbf{E}, \mathcal{K}, \mathbb{A}_{X_i})$

Input: An irreversible system $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and a generic action $\mathbb{A}_{X_i} \triangleq \mathbf{Act}(\mathbb{E}, \mathbb{E}_{\mathbf{pre}}, \mathbb{E}_{\mathbf{add}}, \mathbb{E}_{\mathbf{del}})$ in $\mathcal{K}$.

Output: `true`: $\mathbb{A}_{X_i}$ is instantiated action into $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$; or `false`.

1. $\mathbb{E} := \mathbf{E}$; $\mathbf{E}_{\mathbf{pre}} := \mathbb{E}_{\mathbf{pre}}$; $\mathbf{E}_{\mathbf{add}} := \mathbb{E}_{\mathbf{add}}$; $\mathbf{E}_{\mathbf{del}} := \mathbb{E}_{\mathbf{del}}$;
2. **if** ($\mathbf{E}_{\mathbf{pre}}$ is not `true` in $\mathbf{E}$) **return** `false`; **end if**
3. Apply COA_{BGM} on $\mathbf{E}$ and generate the corresponding graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$;
4. Let $\mathbf{O} \subset \mathbf{N}$ be the set of all observed variables in $\mathbf{Vars}(\mathbf{E})$;
5. **if** ($X_i \notin \mathbf{Anc}(\mathbf{O})$) **return** `true` with $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$; **end if**
6. $\mathbf{P} := \emptyset$;
7. **for each** $N_j \in \mathbf{N}$ where $N_j \neq X_i$ and $N_j \notin \mathbf{O}$
8. **if** ($N_j \in \mathbf{ExVars}(\mathbf{E})$ and $\exists e_{N_j'}$ in $\mathcal{K}$) $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$; $\mathbf{P} := \mathbf{P} \cup N_j$;
9. **else** Create $e_{N_j'}$ as the next time slice of e_{N_j} in $\langle e_{N_j}, N_j \rangle$; $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$;
10. **end for each**
11. **for each** $N_j \in \mathbf{O}$
12. **if** ($N_j \notin \mathbf{Des}(X_i)$) and no ($\exists P_i \in \mathbf{Anc}(N_j)$ and $P_i \in \mathbf{P}$)
13. Create $e_{N_j'}$ as the next time slice of e_{N_j} in $\langle e_{N_j}, N_j \rangle$; $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$; **end if**
14. **end for each**
15. Find $\mathbf{D} \subset \mathbf{N}$ where $\mathbf{D} \not\subseteq \mathbf{Anc}(X_i)$ and $Independent(\mathbf{D}, \mathbf{O} | X_i)$;
16. **for each** $D_j \in \mathbf{D}$
17. $\mathbf{E}_{del} := \mathbf{E}_{del} \cup e_{D_j}$ where $e_{D_j} \in \mathbf{E}$ and e_{D_j} is the mapping of D_j in $\langle e_{D_j}, D_j \rangle$.
18. **end for each**
19. **return** `true` with $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$

Figure 5.9. Procedure for instantiate a generic intervening action $\mathbb{A}_{X_i}$ on a system containing irreversible mechanisms $\mathbf{E}$ using knowledge in $\mathcal{K}$.

To represent intervening actions that are applicable in the same domain for different models, I

represent persistence relations and generic intervening actions into a knowledge base. A persistence relation is represented by a structural equation $e_{X_i'}(X_i', \mathbf{X_{pre}}) = 0$ where $\mathbf{X_{pre}}$ is a set of exogenous variables and $X_i \in \mathbf{X_{pre}}$. The persistence relation describes how a subset of variables in the current time slice, $\mathbf{X_{pre}}$, affects the variable X_i' at the next time slice. I represent generic intervening actions in a knowledge base as $\mathbb{A}_{X_i} \triangleq \mathbf{Act}(\mathbb{E}, \mathbb{E}_{\mathbf{pre}}, \mathbb{E}_{\mathbf{add}}, \mathbb{E}_{\mathbf{del}})$ where $\mathbb{A}_{X_i}$ will be instantiated into $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$ when $\mathbb{A}_{X_i}$ is about to be applied on a model $\mathbf{E}$; $\mathbb{E}_{pre}$ is the precondition that can invoke $\mathbb{A}_{X_i}$; $\mathbb{E}_{\mathbf{add}}$ consists of local mechanisms that will be brought about by the action $\mathbb{A}_{X_i}$ and will be augmented into $\mathbf{E}_{\mathbf{add}}$ when the action $\mathbb{A}_{X_i}$ is instantiated on a specific model $\mathbf{E}$; similarly $\mathbb{E}_{\mathbf{del}}$ is initially an empty set and will be instantiated into $\mathbf{E}_{\mathbf{del}}$ when the action $\mathbb{A}_{X_i}$ is instantiated. Please note that $X_i \in \mathbf{Vars}(\mathbb{E})$ should be in $\mathbb{E}_{\mathbf{pre}}$ as default, since the action will be applied to the variable X_i.

In Figure 5.9, I outline the procedure for instantiating a generic action $\mathbb{A}_{X_i}$ on a model $\mathbf{E}$ using the domain knowledge in $\mathcal{K}$. The procedure *InstantiateActionIrrev*$(\mathbf{E}, \mathcal{K}, \mathbb{A}_{X_i})$ takes a generic intervening action $\mathbb{A}_{X_i}$ on a model $\mathbf{E}$ in the domain $\mathcal{K}$ as inputs and outputs an instantiated action A_{X_i} using knowledge given in $\mathcal{K}$. Line 2 checks if the precondition of the generic intervening action $\mathbb{A}_{X_i}$ is satisfied with $\mathbf{E}$. Lines 5 checks if the system needs to be modeled into consequent time slices. In Lines 7-10, mechanisms for the next time slice are added into $\mathbf{E}_{\mathbf{add}}$. In Line 8, mechanisms for persistence relations are added into $\mathbf{E}_{\mathbf{add}}$. In Lines 11-14, valid observations are created into the system in the next time slice. In Lines 16-18, mechanisms that are independent of the decision at the next time slice are added into $\mathbf{E}_{\mathbf{del}}$. The procedure is worst-case polynomial time due to Line 3.

5.7.3 Myopic Search for Opportunities

Given the way of modeling a sequence of non-intervening and intervening actions in previous subsection, I can now address the problem of search for opportunities with non-intervening and intervening actions on systems containing only irreversible mechanisms. Similarly to the *MyopicSFOIntSysIrrevMechs* procedure presented in Section 5.5, I approximate the optimal strategy computation by myopic search where I consider one intervening (or non-intervening) action at a time and select the one with the maximum value of intervention (or value of observation) to perform. I act according to the selected action and perform the myopic search again to select

the next action to perform until there is no action that can improve the expected utility over the predefined threshold of the increase of expected utility. I outline the procedure in Figures 5.10 and 5.11.

Procedure *MyopicSFOMixSysIrrevMechs*$(M_A, \delta^*, \mathcal{K})$

Input: An augmented model M_A, a threshold δ^* for the increase of expected utility, and the knowledge base $\mathcal{K}$ that contains generic intervening actions $\mathbb{A}_{X_i.m}$ for each manipulable variable $X_i.m$, non-intervening actions $O_{X_i.o}$ for each observable variable $X_i.o$, and persistence relations $e'_{X_i.p}$ for each persistent variable $X_i.p$ in the domain.

Output: A sequence of actions $\mathbf{D}$.

1. $\mathbf{D} := \emptyset$; $M^* := M_A$; *Update*(M^*); $\mu^* := \mathsf{MEU}(M^*)$; *found* := **false**.
2. **while** *found* = **false**
3. $M := M^*$; $\mu := \mu^*$; $\delta := \delta^*$; /* $M = \langle \mathbf{X}, \mathbf{E} \rangle$ */
4. **for each** $X_i \in \mathbf{X}$
5. **if** $X_i \in \mathbf{X}_i^m$
6. **for each** $\mathbb{A}_{X_i}$ in $\mathcal{K}$ /* intervening action */
7. $A_{X_i} = \mathit{InstantiateActionIrrev}(\mathbf{E}, \mathcal{K}, \mathbb{A}_{X_i})$;
8. $M' := \mathit{Augment}(M^*, A_{X_i})$; *Update*$(M')$; $\mu' := \mathsf{MEU}(M')$;
9. $\triangle\mu := \mu' - \mu^*$. /* $\mathsf{VOINT}(A_{X_i})$ */
10. **if** $\triangle\mu > \delta$ **then**
11. $\delta := \triangle\mu$; $M := M'$; $\mu := \mu'$; $A := A_{X_i}$; $d := \pi(M'_i)$; *found* := **true**;
12. **end if**
13. **end for each**
14. **end if**

Continued on Figure 5.11.

Figure 5.10. Myopic search for opportunities with non-intervening and intervening actions on systems containing irreversible mechanisms. *Update*(M) computes the optimal strategy and maximum expected utility for a model M. *Augment*(M, A_{X_i}) (or *Augment*(M, O_{X_i})) denotes the operation of augmenting the model M with an intervening (or non-intervening) action on X_i. *Instantiate*(M, A) denotes the operation of setting (or observing) the value of X_i in M.

The procedure *MyopicSFOMixSysIrrevMechs* takes an augmented model M_A, a threshold δ^*

Continued from Figure 5.10.

15. **if** $X_i \in \mathbf{X}_i^o$ and no $e_{O_{X_i}} \in \mathbf{E}$
16. **for each** O_{X_i} in $\mathcal{K}$ /* non-intervening actions */
17. $M' := Augment(M^*, O_{X_i})$; $Update(M')$; $\mu' := \mathsf{MEU}(M')$;
18. $\triangle\mu := \mu' - \mu^*$. /* $\mathsf{VOOBS}(O_{X_i})$ */
19. **if** $\triangle\mu > \delta$ **then**
20. $\delta := \triangle\mu$; $M := M'$; $\mu := \mu'$; $A := O_{X_i}$; $d := \pi(M_i')$; $found :=$ **true**;
21. **end if**
22. **end for each**
23. **end if**
24. **end for each**
25. **if** $found =$ **true then**
26. $M^* := Instantiate(M, A)$; $\mu^* := \mu$; $\mathbf{D} := \mathbf{D} \cup \{A\}$;
27. $found :=$ **false**,
28. **else** $found :=$ **true**.
29. **end if**
30. **end while**.
31. **return D**.

Figure 5.11. Myopic search for opportunities with non-intervening and intervening actions on systems containing irreversible mechanisms (continued).

for the increase of expected utility, and the domain knowledge base $\mathcal{K}$ as inputs and outputs a sequence of actions $\mathbf{D}$. The domain knowledge base $\mathcal{K}$ contains generic intervening actions $\mathbb{A}_{X_i.m}$ for each manipulable variable $X_i.m$, non-intervening actions $O_{X_i.o}$ for each observable variable $X_i.o$, and persistence relations $e'_{X_i.p}$ for each persistent variable $X_i.p$ in the domain. In other words, in addition to the manipulability, observability, and focus, I introduce the *persistence* property for each variable and specify their persistent relations in the knowledge base. In Lines 4-24, the procedure performs the myopic search to find the best intervening actions or non-intervening action to perform. In Lines 6-13, the procedure goes through each manipulable variable $X_i.m$ and its

admissible intervening actions to compute the value of interventions for the myopic search. In Lines 16-22, the procedure goes through each unobserved observable variable $X_i.o$ to compute the value of observation for the myopic search. The complexity of this procedure is NP-hard since it depends on Bayesian network inference in Lines 8 and 17.

5.8 Search for Opportunities in Systems Containing Reversible Mechanisms

In Chapter 4, I show that performing a self-contained action on a system containing reversible mechanisms may reverse causal relations among variables in the system. I presented algorithms *FindAtomicDeletions* and *FindAtomicAdditions* in Section 4.4 to assist atomic action deliberation in systems containing reversible mechanisms. In previous sections, I addressed the problem of search for opportunities, using recursive actions defined in Definition 5.1, for systems consisting of irreversible mechanisms. In this section, I address the problem of search for opportunities, using self-contained actions defined in Definition 4.5, for systems containing reversible mechanisms.

First, I consider the case where I apply only intervening actions on systems containing reversible mechanisms. For each manipulable variable $X_i.m$ in the system, I can use *FindAtomicDeletions* procedure in Figure 4.2 to identify all possible atomic actions for manipulating $X_i.m$. For each endogenous variable X_i in the system, I can use *FindAtomicAdditions* procedure in Figure 4.4 to identify all possible atomic actions which one can consider in order to release the mechanism governing X_i. I can then compute the value of intervention for applying each of these identified atomic actions on the system, and then select the one that yields the maximum increase of the expected utility as the next action to perform. I outline this procedure in Figures 5.12 and 5.13.

The procedure *MyopicSFOIntSysRevMechs*$(M_A, \delta^*, \mathcal{K})$ takes the augmented model M_A, a threshold δ^* for the increase of expected utility, and the knowledge base $\mathcal{K}$ that contains the manipulability and reversibility of domain variables and outputs a sequence of actions $\mathbf{D}$. In Lines 4-14, the procedure loops through each manipulable variable in the model to identify their possible atomic deletions and to generate possible atomic actions for the value of intervention computation in Line 8. In Lines 15-26, the procedure loops through each endogenous variable in the model to identify their

Procedure *MyopicSFOIntSysRevMechs*$(M_A, \delta^*, \mathcal{K})$

Input: An augmented model M_A, a threshold δ^* for the increase of expected utility, and the knowledge base $\mathcal{K}$ that contains the manipulability and reversibility of domain variables.

Output: A sequence of actions $\mathbf{D}$.

1. $\mathbf{D} := \emptyset$; $M^* := M_A$; *Update*(M^*); $\mu^* := \mathsf{MEU}(M^*)$; *found* := **false**.
2. **while** *found* = **false**
3. $M := M^*$; $\mu := \mu^*$; $\delta := \delta^*$; /* $M = \langle \mathbf{X}, \mathbf{E} \rangle$ */
4. **for each** $X_i \in \mathbf{X}_i^m$ where $\mathbf{X}_i^m \subseteq \mathbf{X}$
5. $\mathbf{E}_{\mathbf{del}} := \mathit{FindAtomicDeletions}(\mathbf{E}, \mathcal{K}, \mathbf{E}_{add(X_i)})$
6. **for each** $\mathbf{E}_{del(d)} \in \mathbf{E}_{\mathbf{del}}$
7. $A_{X_i} := \mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E}_{add(X_i)}, \mathbf{E}_{del(d)})$
8. $M' := \mathit{Augment}(M^*, A_{X_i})$; *Update*$(M')$; $\mu' := \mathsf{MEU}(M')$;
9. $\triangle\mu := \mu' - \mu^*$. /* $\mathsf{VOINT}(A_{X_i})$ */
10. **if** $\triangle\mu > \delta$ **then**
11. $\delta := \triangle\mu$; $M := M'$; $\mu := \mu'$; $A := A_{X_i}$; $d := \pi(M'_i)$; *found* := **true**;
12. **end if**
13. **end for each**
14. **end for each**

Continued on Figure 5.13.

Figure 5.12. Myopic search for opportunities with intervening actions on systems containing reversible mechanisms. *Update*(M) computes the optimal strategy and maximum expected utility for a model M. *Augment*(M, A_{X_i}) denotes the operation of augmenting the model M with an intervening action on X_i. *Instantiate*(M, A_{X_i}) denotes the operation of setting the value of X_i in M.

possible atomic additions and to generate possible atomic actions for the value of intervention computation in Line 20. The complexity of this procedure is also NP-hard due to the belief updates in Lines 8 and 20.

Now, I consider applying a sequence of intervening and non-intervening actions on systems containing reversible mechanisms. The method for modeling a sequence of non-intervening and intervening actions on systems containing reversible mechanisms is similar to the one modeling

Continued from Figure 5.12.

15. **for each** $X_j \in \mathbf{EnVars}(E)$
16. **find** e in the mapping $\langle X_j, e\rangle$ in $G(E)$;
17. $\mathbf{E_{add}} := \mathit{FindAtomicAdditions}(\mathbf{E}, \mathcal{K}, \mathbf{E}_{del(e)})$
18. **for each** $\mathbf{E}_{add(X_i)} \in \mathbf{E_{add}}$
19. $A_{X_i} := \mathbf{Act}(\mathbf{E}, \emptyset, \mathbf{E}_{add(X_i)}, \mathbf{E}_{del(e)})$
20. $M' := \mathit{Augment}(M^*, A_{X_i})$; $\mathit{Update}(M')$; $\mu' := \mathsf{MEU}(M')$;
21. $\triangle\mu := \mu' - \mu^*$. /* $\mathsf{VOINT}(A_{X_i})$ */
22. **if** $\triangle\mu > \delta$ **then**
23. $\delta := \triangle\mu$; $M := M'$; $\mu := \mu'$; $A := A_{X_i}$; $d := \pi(M'_i)$; $\mathit{found} :=$ **true**;
24. **end if**
25. **end for each**
26. **end for each**
27. **if** $\mathit{found} =$ **true then**
28. $M^* := \mathit{Instantiate}(M, A)$; $\mu^* := \mu$; $\mathbf{D} := \mathbf{D} \cup \{A\}$; $\mathit{found} :=$ **false**;
29. **else** $\mathit{found} :=$ **true**;
30. **end if**
31. **end while**

Figure 5.13. Myopic search for opportunities with intervening actions on systems containing reversible mechanisms (continued).

actions on systems containing only irreversible mechanisms, except that the criteria for inferring the persistence and response need to be modified for actions on reversible mechanisms. With respect to persistence relations, I use the same criterion to apply persistence relations on exogenous variables in the next time slice, except for the manipulated variable. With respect to the response, I shall create the system in the next time slice as applying the action on the system in the current time slice. And I shall drop invalid observations, which are descendants of the manipulated variable in the system of the next time slices and descendants of exogenous variables which are involved in persistence relations, and keep the valid ones. In addition, I shall also drop those

mechanisms that are not needed for decisions at the next time slice from the system in the current time slice. Consider the system depicted in Figure 5.14(a) that describes relations among variables A, B, C, D, E, F, M, N with non-intervening actions $D_{O_n}, D_{O_d}, D_{O_e}$ and their corresponding observations O_n, O_d, O_e. For the sake of presentation, I do not include mechanisms describing utility functions in the system. Consider the generic intervening action $\mathbb{A}_E \triangleq \mathbf{Act}(\mathbb{E}, \mathbb{E}_{\mathbf{pre}}, \mathbb{E}_{\mathbf{add}}, \mathbb{E}_{\mathbf{del}})$ where $\mathbb{E}_{\mathbf{pre}} = \{O_e \in \mathbf{Vars}(\mathbf{E}) \wedge D_{O_e} = \texttt{true}\}$, $\mathbb{E}_{\mathbf{add}} = \{f_E\}$, $\mathbb{E}_{\mathbf{del}} = \{f_C\}$. When I instantiate this generic action on the model $\mathbf{E}$ depicted in Figure 5.14(a), I add $f_{A'}$ into $\mathbf{E}_{\mathbf{add}}$ by persistence relations in $\mathcal{K}$. With respect to the response, I add $f_{M'}, f_{N'}, f_{B'}, f_{D'}, f_{F'}, f_{O'_n}, f_{D'_{O_n}}$ into $\mathbf{E}_{\mathbf{add}}$ by copying them directly from their corresponding mechanisms in previous time slice. I copy f_E into $\mathbf{E}_{\mathbf{add}}$ as $f_{C'}$ and do not copy f_C into $\mathbf{E}_{\mathbf{add}}$. And f_E specified in $\mathbb{E}_{\mathbf{add}}$ is instantiated into $f_{E'}$ is added into $\mathbf{E}_{\mathbf{add}}$ together with f_{D_e} .

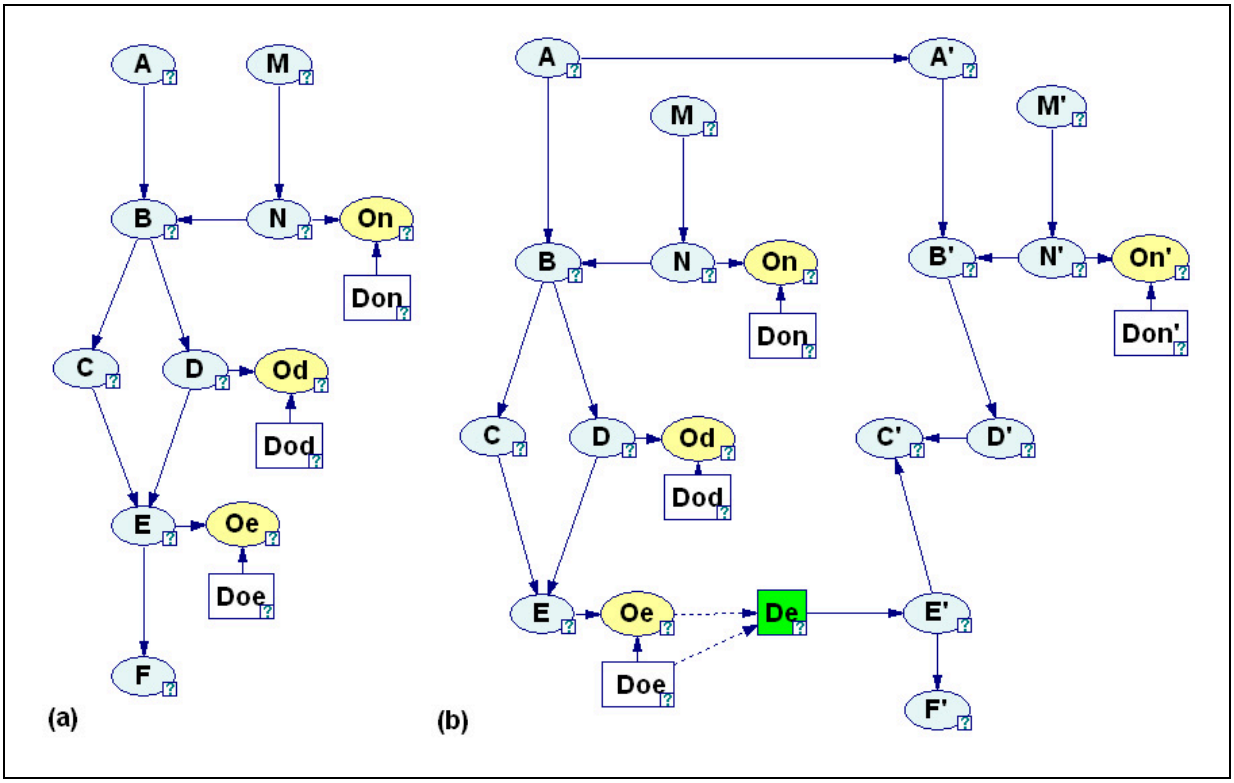


Figure 5.14. (a) depicts relations among variables A, B, C, D, E, F, M, N and non-intervening actions $D_{O_n}, D_{O_d}, D_{O_e}$ and their corresponding observations O_n, O_d, O_e. (b) depicts the augmented model for considering the intervening action which manipulates on E and releases the mechanism governing C. The system in (a) is augmented into the next time slice in (b). There is a persistence relation for A defined in knowledge base, but no persistence relation for M. The valid observation O_n is therefore kept in the next time slice.

Procedure *InstantiateActionRev*$(\mathbf{E}, \mathcal{K}, \mathbb{A}_{X_i})$

Input: A reversible system $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and a generic action $\mathbb{A}_{X_i} \triangleq \mathbf{Act}(\mathbb{E}, \mathbb{E}_{\mathbf{pre}}, \mathbb{E}_{\mathbf{add}}, \mathbb{E}_{\mathbf{del}})$ in $\mathcal{K}$.

Output: `true`: $\mathbb{A}_{X_i}$ is instantiated action into $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$; or `false`.

1. $\mathbb{E} := \mathbf{E}$; $\mathbf{E}_{\mathbf{pre}} := \mathbb{E}_{\mathbf{pre}}$; $\mathbf{E}_{\mathbf{add}} := \emptyset$; $\mathbf{E}_{\mathbf{del}} := \emptyset$;
2. **if** ($\mathbf{E}_{\mathbf{pre}}$ is not `true` in $\mathbf{E}$) **return** `false`; **end if**
3. Apply COA_{BGM} on $\mathbf{E}$ and generate the corresponding graph $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A} \rangle$;
4. Let $\mathbf{O} \subset \mathbf{N}$ be the set of all observed variables in $\mathbf{Vars}(\mathbf{E})$;
5. $\mathbf{P} := \emptyset$;
6. **for each** $N_j \in \mathbf{N}$ where $N_j \neq X_i$ and $N_j \notin \mathbf{O}$
7. **if** ($N_j \in \mathbf{ExVars}(\mathbf{E})$ and $\exists e_{N_j'}$ in $\mathcal{K}$) $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$; $\mathbf{P} := \mathbf{P} \cup N_j$;
8. **else if** ($N_j = \mathbf{Vars}(\mathbb{E}_{\mathbf{add}})$)
9. Copy e_{N_j} into $e_{N_j'}$ where $e_{N_j} \in \mathbb{E}_{\mathbf{add}}$; $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$;
10. **else if** ($N_j \neq N_i$ where $e_{N_i} \in \mathbb{E}_{\mathbf{del}}$)
11. Copy e_{N_j} into $e_{N_j'}$ where e_{N_j} in $\langle e_{N_j,N_j} \rangle$; $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$; **end if**
12. **end for each**
13. **for each** ($N_j \in \mathbf{O}$)
14. **if** ($N_j \notin \mathbf{Des}(X_i)$) and no ($\exists P_i \in \mathbf{Anc}(N_j)$ and $P_i \in \mathbf{P}$)
15. Copy e_{N_j} into $e_{N_j'}$ where e_{N_j} in $\langle e_{N_j,N_j} \rangle$; $\mathbf{E}_{add} := \mathbf{E}_{add} \cup e_{N_j'}$; **end if**
16. **end for each**
17. Find $\mathbf{D} \subset \mathbf{N}$ where $\mathbf{D} \not\subseteq \mathbf{Anc}(X_i)$ and $\mathit{Independent}(\mathbf{D}, \mathbf{O} | X_i)$;
18. **for each** $D_j \in \mathbf{D}$
19. $\mathbf{E}_{del} := \mathbf{E}_{del} \cup e_{D_j}$ where $e_{D_j} \in \mathbf{E}$ and e_{D_j} is the mapping of D_j in $\langle e_{D_j}, D_j \rangle$.
20. **end for each**
21. **return** `true` with $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$

Figure 5.15. Procedure for instantiating a generic intervening action $\mathbb{A}_{X_i}$ on a system containing reversible mechanisms $\mathbf{E}$ using knowledge in $\mathcal{K}$.

I outline the procedure for instantiating generic intervening actions on systems containing reversible mechanisms in Figure 5.15. The procedure *InstantiateActionRev* takes a reversible system $\mathbf{E}$, a mechanism knowledge base $\mathcal{K}$, and a generic intervening action $\mathbb{A}_{X_i} \triangleq \mathbf{Act}(\mathbb{E}, \mathbb{E}_{\mathbf{pre}}, \mathbb{E}_{\mathbf{add}}, \mathbb{E}_{\mathbf{del}})$ in $\mathcal{K}$ as inputs and outputs an instantiated action $A_{X_i} \triangleq \mathbf{Act}(\mathbf{E}, \mathbf{E}_{\mathbf{pre}}, \mathbf{E}_{\mathbf{add}}, \mathbf{E}_{\mathbf{del}})$ when the precondition of $\mathbb{A}_{X_i}$ is satisfied. In Lines 6-12, mechanisms for the next time slice are added into $\mathbf{E}_{\mathbf{add}}$. In Line 7, mechanisms for persistence relations are added into $\mathbf{E}_{\mathbf{add}}$. In Lines 13-16, valid observations are created into the system in the next time slice. In Lines 18-20, mechanisms that are independent to the decision at the next time slice are added into $\mathbf{E}_{\mathbf{del}}$. The procedure is worst-case polynomial time due to Line 3.

Given *InstantiateActionRev* outlined in Figure 5.15, I present the myopic approach of search for opportunities in systems containing reversible mechanisms in Figure 5.16 and 5.17. The procedure $MyopicSFO(M_A, \delta^*, \mathcal{K})$ takes an augmented model M_A, a threshold δ^* for the increase of expected utility, and the knowledge base $\mathcal{K}$ as inputs and produces a sequence of actions. Lines 1-13 adds generic actions found by $FindAtomicDeletions$ and $FindAtomicDeletions$ into $\mathcal{K}$. Lines 17-37 performs the myopic search. Lines 19-26 searches through intervening actions in $\mathcal{K}$ and compute their values of interventions. Lines 29-35 search through non-intervening actions in $\mathcal{K}$ and compute their values of observations.

5.9 Discussion

In this chapter, I presented augmented causal models to address the problem of search for opportunities. I introduced the myopic search algorithms that compute the value of intervention and the value of observation to myopically select the best action to perform. The proposed myopic search algorithms can work with systems containing mixtures of mechanisms. Table 5.2 presents a categorization of proposed algorithms for myopic search for opportunities along the reversibility of systems and the availability of observations. The algorithms for reversible systems subsume the algorithms for irreversible systems, as do the algorithms for observable systems to unobservable systems.

The concept of value of intervention has also been proposed for causal discovery in active learning [Murphy, 2001; Tong and Koller, 2001; Yoo and Cooper, 2002]. Since the focus of their

Procedure *MyopicSFO*$(M_A, \delta^*, \mathcal{K})$

Input: An augmented model M_A, a threshold δ^* for the increase of expected utility, and the knowledge base $\mathcal{K}$ that contains generic intervening actions $\mathbb{A}_{X_i.m}$ for each manipulable variable $X_i.m$, non-intervening actions $O_{X_i.o}$ for each observable variable $X_i.o$, and persistence relations $e'_{X_i.p}$ for each persistent variable $X_i.p$ in the domain.

Output: A sequence of actions $\mathbf{D}$.

1. **for each** $X_i \in \mathbf{X}_i^m$
2. $\quad \mathbf{D_{del}} := \mathit{FindAtomicDeletions}(\mathbf{E}, \mathcal{K}, \mathbf{E}_{add(X_i)})$
3. $\quad$ **for each** $D_j \in \mathbf{D_{del}}$
4. $\quad\quad$ Add generic intervening action $\mathbb{A}_{X_i} = \mathbf{Act}(\mathbb{E}, \emptyset, \mathbb{E}_{add(X_i)}, \mathbb{E}_{del(e_{D_j})})$ into $\mathcal{K}$.
5. $\quad$ **end for each**
6. **end for each**
7. **for each** $X_i \in \mathbf{EnVars}(\mathbf{E})$
8. $\quad$ Let e_{X_i} be the mapping of X_i in $G(E)$.
9. $\quad \mathbf{D_{add}} := \mathit{FindAtomicAdditions}(\mathbf{E}, \mathcal{K}, \mathbf{E}_{del(e_{X_i})})$ into $\mathcal{K}$.
10. $\quad$ **for each** $D_j \in \mathbf{D_{add}}$
11. $\quad\quad$ Add generic intervening action $\mathbb{A}_{D_j} = \mathbf{Act}(\mathbb{E}, \emptyset, \mathbb{E}_{add(D_j)}, \mathbb{E}_{del(e_{X_i})})$
12. $\quad$ **end for each**
13. **end for each**
14. $\mathbf{D} := \emptyset$; $M^* := M_A$; $\mathit{Update}(M^*)$; $\mu^* := \mathsf{MEU}(M^*)$; $\mathit{found} :=$ **false**.
15. **while** $\mathit{found} =$ **false**
16. $\quad M := M^*$; $\mu := \mu^*$; $\delta := \delta^*$; /* $M = \langle \mathbf{X}, \mathbf{E} \rangle$ */
17. $\quad$ **for each** $X_i \in \mathbf{X}$
18. $\quad\quad$ **if** $X_i \in \mathbf{X}_i^m$

Continued on Figure 5.17.

Figure 5.16. Myopic search for opportunities in systems containing reversible mechanisms. $\mathit{Update}(M)$ computes the optimal strategy and maximum expected utility for a model M. $\mathit{Augment}(M, A_{X_i})$ (or $\mathit{Augment}(M, O_{X_i})$) denotes the operation of augmenting the model M with an intervening (or non-intervening) action on X_i. $\mathit{Instantiate}(M, A)$ denotes the operation of setting (or observing) the value of X_i in M.

Continued from Figure 5.16.

```
19.          for each 𝔸_{X_i} in 𝒦 /* intervening action */
20.              A_{X_i} = InstantiateActionRev(E, 𝒦, 𝔸_{X_i});
21.              M' := Augment(M*, A_{X_i}); Update(M'); μ' := MEU(M');
22.              △μ := μ' − μ*. /* VOINT(A_{X_i}) */
23.              if △μ > δ then
24.                  δ := △μ; M := M'; μ := μ'; A := A_{X_i}; d := π(M'_i); found := true;
25.              end if
26.          end for each
27.      end if
28.      if X_i ∈ X_i^o and no e_{O_{X_i}} ∈ E
29.          for each O_{X_i} in 𝒦 /* non-intervening actions */
30.              M' := Augment(M*, O_{X_i}); Update(M'); μ' := MEU(M');
31.              △μ := μ' − μ*. /* VOOBS(O_{X_i}) */
32.              if △μ > δ then
33.                  δ := △μ; M := M'; μ := μ'; A := O_{X_i}; d := π(M'_i); found := true;
34.              end if
35.          end for each
36.      end if
37.  end for each
38.  if found = true then
39.      M* := Instantiate(M, A); μ* := μ; D := D ∪ {A}; found := false,
40.  else found := true.
41.  end if
42. end while.
43. return D.
```

Figure 5.17. Myopic search for opportunities in systems containing reversible mechanisms (continued).

Table 5.2. A categorization of algorithms for myopic search for opportunities.

Systems	unobservable	observable
irreversible	*MyopicSFOIntSysIrrevMechs*	*MyopicSFOMixSysIrrevMechs*
reversible	*MyopicSFOIntSysRevMechs*	*MyopicSFO*

work is on discovering the true model structure, the value of intervention is defined over all possible models. My approach, on the other hand, assumes the availability of the true model and uses the value of intervention to advise the next intervention to perform to achieve a desired objective.

Jensen and Vomlelová [2002] introduced *unconstrained influence diagrams* that address decision problems in which the order of decisions and observations is not determined, but partial temporal ordering of decisions and observations is specified. In my framework, I address decision problems where the choice of a variable being observed or intervened is not even determined. The myopic search for opportunities procedure suggests users on which variables to observe and on which variables to intervene.

Breese and Heckerman [1996] introduced the persistence network for modeling the repair in decision-theoretic troubleshooting. Their persistence network is similar to the modeling of the persistence and response relations discussed in this chapter. However, my approach can handle systems with mixture of mechanisms and the mechanism-based view of causality allows us to infer the persistence and response relations with respect to generic actions.

Chapter 6
Evaluation

Chapter 3 proposes the mechanism-based causal model construction to assist users in constructing causal models. Built on the results of theoretical analysis, the proposed framework provides a sound and novel approach to model construction. However, the soundness and novelty alone do not imply the effectiveness of the proposed framework [Suermondt, 1992]. This chapter describes a preliminary study in determining the plausibility that *ImaGeNIe* provides an effective environment in constructing causal models. I investigated the effectiveness of *ImaGeNIe* in the domain of stock market investment, measuring (1) the structure discrepancy between constructed models and gold standard models, and (2) the efficiency of model constructions. I also recorded users' rating of the usefulness of *ImaGeNIe*.

Section 6.1 reports my method of studying the effectiveness of using *ImaGeNIe* in causal model constructions and changes in structure. Section 6.2 presents the experimental results. Section 6.3 summarizes the findings and discusses the possible threats to my study.

6.1 Methods

To evaluate the effectiveness of using *ImaGeNIe* in causal model constructions, I presented the subjects with the problem descriptions (cases) generated from gold standard models; and asked them to construct model structures for each case by means of *ImaGeNIe*, *EqGeNIe* and *Excel*.[1] I measured the effectiveness of *ImaGeNIe* by comparing the structure discrepancies between the models constructed by *ImaGeNIe* and *EqGeNIe* to gold standard models. Finally, I measured the efficiency by comparing the time spent on constructing models using *ImaGeNIe*, *EqGeNIe* and *Excel*.

The study has a within-subjects case-by-case experimental design. Each problem description is a case. For each case, subjects use *ImaGeNIe*, *EqGeNIe* and *Excel* to construct models. One unit of measure is the structure discrepancy between the constructed model and the gold standard

[1]Please see Section 6.1.2 for a detail description of *EqGeNIe* system.

model of a single case by a single subject; and the other unit of measure is the time spent on a single case by a single subject. At the end of the experiment, subjects were asked to fill a questionnaire to rate the usefulness of *ImaGeNIe*, *EqGeNIe* and *Excel*.

Section 6.1.1 describes the problem domain, the selection of gold standard models, and the generation of cases. Section 6.1.2 describes the presentation of *ImaGeNIe*, *EqGeNIe* and *Excel* systems. Section 6.1.3 describes the issues of subjects, randomization, procedures, and questionnaire in the experimental design. Section 6.1.4 describes the proposed measure of structure discrepancy and time, and the statistical analysis of results.

6.1.1 Case Generation

The domain of this evaluation is stock market investment. Please see Appendix B.1 for a detailed description of the simplified mechanism library for stock market investment. The reason for choosing this domain is its accessibility and familiarity to the subjects. I chose four mechanisms which consisted of nine variables from the domain to generate two cases for the study. The first case (Task I) asked subjects to answer a query by constructing models using *ImaGeNIe*, *EqGeNIe* and *Excel*. The second case (Task II) asked subjects to answer another query by changing the structures of constructed models of Task I using *ImaGeNIe*, *EqGeNIe* and *Excel* . For each case, I generated a case description. Please see Appendix B.2 for the presentation of the case descriptions. I also generated a gold standard model for each case. I ensured that the gold standard models of the two cases are directed acyclic graphs, i.e., they contained no strongly-coupled components. The main reason for such restriction is that existing systems are only designed for constructing acyclic graphical models such as Bayesian networks or influence diagrams. Although *ImaGeNIe* supports strongly-coupled components, I restricted my study to cases represented by directed acyclic graphs so that I can compare *ImaGeNIe* with *EqGeNIe* and *Excel* systems.

6.1.2 System Presentations

Since *ImaGeNIe* is designed as an embedded model construction module in *GeNIe*, the graphical user interfaces of *ImaGeNIe* and *GeNIe* are basically consistent. However, they can be easily distinguished from each other since *ImaGeNIe* has an additional mechanism-tree view window and

mechanism boxes in the workspace. To ensure that subjects have clear understanding of different approaches (mechanism-based versus traditional) in the experiment, it was emphasized that the GUI with mechanism-tree view window is called *ImaGeNIe* and the traditional *GeNIe* workspace is called *EqGeNIe* since it contains an additional equation node type. Figure 6.1 shows *EqGeNIe* system where subjects can draw directed arcs among equation nodes and enter equations with the equation authoring dialogue. For each case, each subject used *ImaGeNIe*, *EqGeNIe* and *Excel* to construct models. Please note that I did not ask the subjects to use *Excel* to construct graphical models but rather use *Excel* to solve the case in the spreadsheet view of *Excel*, i.e., using the formula bar in *Excel* to enter equations. I also pre-created the variables participating in the tasks in the workspace of *EqGeNIe* and the spreadsheet of *Excel*. This ensured that *EqGeNIe* and *ImaGeNIe* could be compared on an equal footing, since variables and mechanisms are stored in the stock market investment mechanism library for *ImaGeNIe*.

To ensure that subjects are familiar with the functionalities needed for solving cases in *ImaGeNIe*, *EqGeNIe* and *Excel*, I showed the available systems in the modeling environments and demonstrated their uses in constructing models for an example case. Then I assisted each subject in constructing models for the example case and made sure that each subject had sufficient confidence and skill in working with *ImaGeNIe*, *EqGeNIe* and *Excel*. Please see Appendix B.3 for the presentation of the example case and the training session.

6.1.3 Design

This study had a within-subjects case-by-case experimental design, which allowed me to cope with the constraint of limited number of subjects participating in both control and experimental setting. In this section, I will introduce the background of the subjects, the procedures, the control of potential sources of bias, and the content of the questionnaire.

Subjects: 40 subjects participated in the experiments. All subjects were graduate students taking the class Decision Analysis and Decision Support Systems (INFSCI 2130 / ISSP 2240) offered by Prof. Marek J. Druzdzel in the Spring 2003 semester at the School of Information Sciences or graduate students, members of Decision Systems Laboratory who have taken this class in the past. The request for voluntary subjects was announced through the course e-mail list.

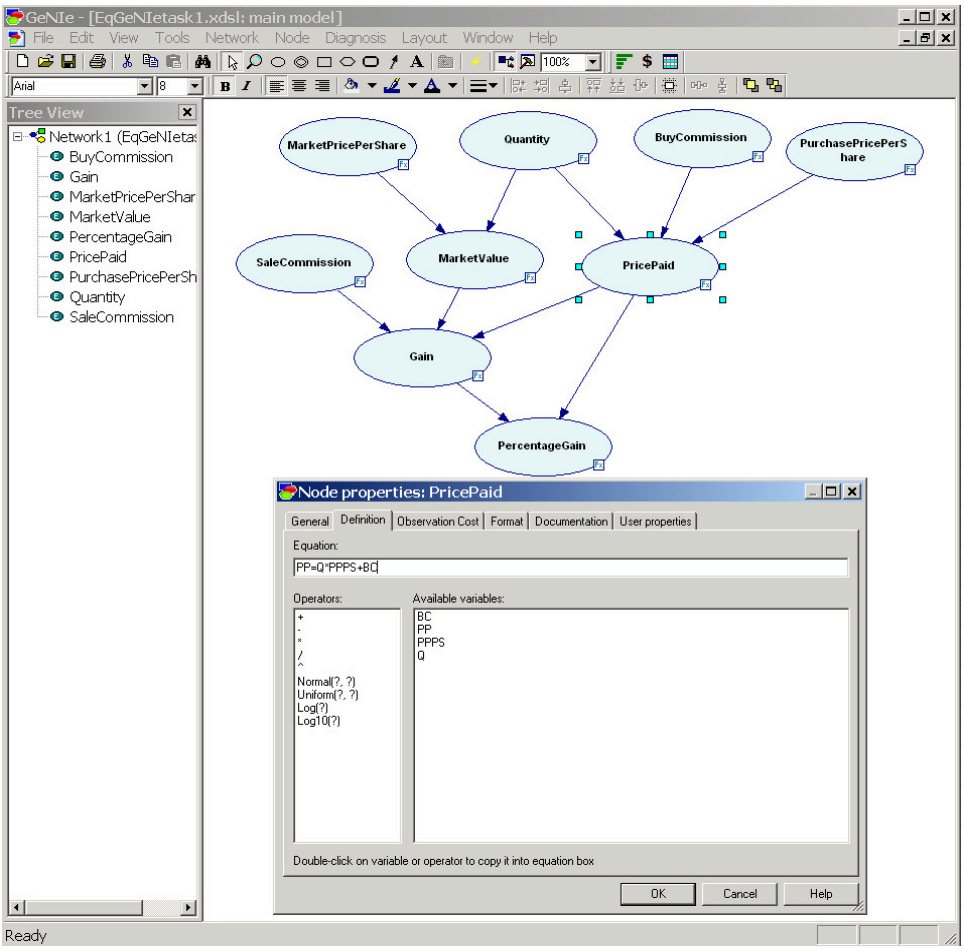


Figure 6.1. *EqGeNIe* System. The equation nodes are shown as green "E" icons in the network tree view window at left. Users define equations for equation nodes with equation authoring dialogue shown at the bottom.

Subjects were compensated for their participation by extra course credit and candies. All subjects have received basic training in decision analysis and were familiar with the graphical decision support tools such as *GeNIe*, as they had been using *GeNIe* in solving exercises for the class.

Procedures: Each subject received the following procedures:

1. A training session in using *ImaGeNIe*, *EqGeNIe* and *Excel* to construct models for the example case.
2. Two cases to solve using *ImaGeNIe*, *EqGeNIe* and *Excel* in a randomized order to

construct models for each case.

3. A questionnaire to fill out after the experiment.

Control of potential bias: 1. To control for a possible bias generated by the order in which the training session of using *EqGeNIe*, *ImaGeNIe* and *Excel* were given to the subjects, I randomly assigned for each subject which training session the subject would receive first.

2. To control for a possible bias generated by the order in which the systems *ImaGeNIe*, *EqGeNIe* and *Excel* are used to construct models for each case, I randomized this order for each subject.

Questionnaire: I ask the subjects to rate the usefulness of the three systems for solving each task case on an eleven-point scale ranged from *useless* (0) to *extremely useful* (10). I then asked an open-ended question for subjects' comments on using different systems in solving each task case. Please see Appendix B.4 for the presentation of the questionnaire.

6.1.4 Data Analysis

Effectiveness

Of the 40 subjects who participated in the experiment, two dropped out the experiments for personal reasons, leaving 38 usable results. I further eliminated the last two subjects to counterbalance the order of using *ImaGeNIe*, *EqGeNIe* and *Excel*.

The unit of one of the measures in this study was the structure discrepancy between the constructed model and the gold standard model of a single case by a single subject. Although models constructed by *Excel* system were not in a graphical form, I converted them into directed graphs by reading off the dependency relations specified in cells. For example, if a cell "$C3$" in a spreadsheet view had formula "$= C1 + C2$", I drew directed arcs $C1 \rightarrow C3$ and $C2 \rightarrow C3$. With three systems, thirty-six subjects, and two tasks, I have $3 \times 36 \times 2$ data points for statistical analysis.

Since I need to process models in a graphical form, I first discuss how to read off a model structure from a causal graph. Given a causal graph G over variables $\mathbf{V} = \{V_1, \ldots, V_n\}$, I create a n by n matrix M where columns and rows are indexed by the the same order of $\mathbf{V}$ and all its elements are 0. For each node $V_i \in \mathbf{V}$ and its parents $\mathbf{Pa}(V_i) \subset \mathbf{V}$ in G, I change the elements a_{ii}

and a_{ij}, where column $V_j \in \mathbf{Pa}(V_i)$, to 1. This change corresponds to reading off G the structural equations and the variables participating in each equation. The constructed matrix is called an *ordered binary structure matrix* of G, since it has binary values as its elements and a fixed order of variables in rows and columns.

Given two causal graph G and G' over the same set of variable $\mathbf{V}$, I define the *structure similarity measure* as follows.

Definition 6.1 (structure similarity measure) *Let G and G' be two causal graphs over the same set of variables $\mathbf{V}$. Let M and M' be the ordered binary structure matrices of G and G' respectively, where M and M' have the same order over $\mathbf{V}$. I define* structure similarity measure *between M and M' as $D(M, M') \triangleq \sum_{ij} a_{ij} \oplus a'_{ij}$, where a_{ij} and a'_{ij} are the elements of M and M' respectively, and $\oplus$ is the exclusive-or operator.*

The structure similarity measure satisfies three fundamental properties of distance and, hence, it is a *distance measure*.

Theorem 6.1

Let G, G', and G'' be causal graphs over the same set of variables $\mathbf{V}$. Let M, M', M'' be the ordered binary structure matrices of G G', and G'' respectively, where M, M', M'' have the same row and column order over $\mathbf{V}$. Then:

1. Positiveness*: $|\mathbf{V}| \times |\mathbf{V}| \geq D(M, M') \geq 0$. $D(M, M') = 0$ iff $a_{ij} = a'_{ij}$ for all ij, and $D(M, M') = |\mathbf{V}| \times |\mathbf{V}|$ iff $a_{ij} \neq a'_{ij}$ for all ij, where a_{ij} and a'_{ij} are elements of M and M' respectively.*

2. Symmetry*: $D(M, M') = D(M', M)$.*

3. Triangle Inequality*: $D(M, M') + D(M', M'') \geq D(M, M'')$.*

Proof: The proofs for positiveness and symmetry are straightforward. Here, I will only prove the triangle inequality. We first expand the equation

$$D(M, M') + D(M', M'') \geq D(M, M'') \tag{6.1}$$

to

$$\sum_{ij} a_{ij} \oplus a'_{ij} + \sum_{ij} a'_{ij} \oplus a''_{ij} \geq \sum_{ij} a_{ij} \oplus a''_{ij} \tag{6.2}$$

where a_{ij}, a'_{ij}, and a''_{ij} are elements of M, M', and M'' respectively. Since M, M', and M'' have the same dimensions, we move the summation over ij in front of the left hand side of Equation 6.2. We have

$$\sum_{ij} [(a_{ij} \oplus a'_{ij}) + (a'_{ij} \oplus a''_{ij})] \geq \sum_{ij} a_{ij} \oplus a''_{ij}. \tag{6.3}$$

For each ij, if we have $(a_{ij} \oplus a'_{ij}) + (a'_{ij} \oplus a''_{ij}) \geq a_{ij} \oplus a''_{ij}$, then Equation 6.3 is true and so is the triangle inequality. In Table 6.1, we see that for any combinations of a_{ij}, a'_{ij}, and a''_{ij}, the inequality $(a_{ij} \oplus a'_{ij}) + (a'_{ij} \oplus a''_{ij}) \geq a_{ij} \oplus a''_{ij}$ holds. Therefore, we prove that the structure similarity measure satisfies the triangle inequality. □

Table 6.1. Enumerations of a_{ij}, a'_{ij}, and a''_{ij} for the proof of triangle inequality for the structure similarity measure.

a_{ij}	a'_{ij}	a''_{ij}	$a_{ij} \oplus a'_{ij}$	$a'_{ij} \oplus a''_{ij}$	$a_{ij} \oplus a''_{ij}$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	0	0

Example 6.1 Consider the causal graphs (a) and (b) over the same set of variables $\{A, B, C, D, E\}$ in Figure 6.2. Their ordered binary structure matrices is shown in Figure 6.3, and their distance is 6. □

For each case c_i, I denote its gold standard model as $M_g(c_i)$. For each subject s_j, I denote the models constructed for each case c_i using *ImaGeNIe*, *EqGeNIe* and *Excel* as $M_{Ima}(c_i, s_j)$ and $M_{Eq}(c_i, s_j)$ and $M_{Ex}(c_i, s_j)$ respectively. Given that all $M_g(c_i)$, $M_{Ima}(c_i, s_j)$, $M_{Eq}(c_i, s_j)$, and $M_{Ex}(c_i, s_j)$ have the same column and row order over $\mathbf{V}$, for each case c_i and each subject s_j I have three data points $D(M_g(c_i), M_{Ima}(c_i, s_j))$, $D(M_g(c_i), M_{Eq}(c_i, s_j))$, and $D(M_g(c_i), M_{Ex}(c_i, s_j))$. Each

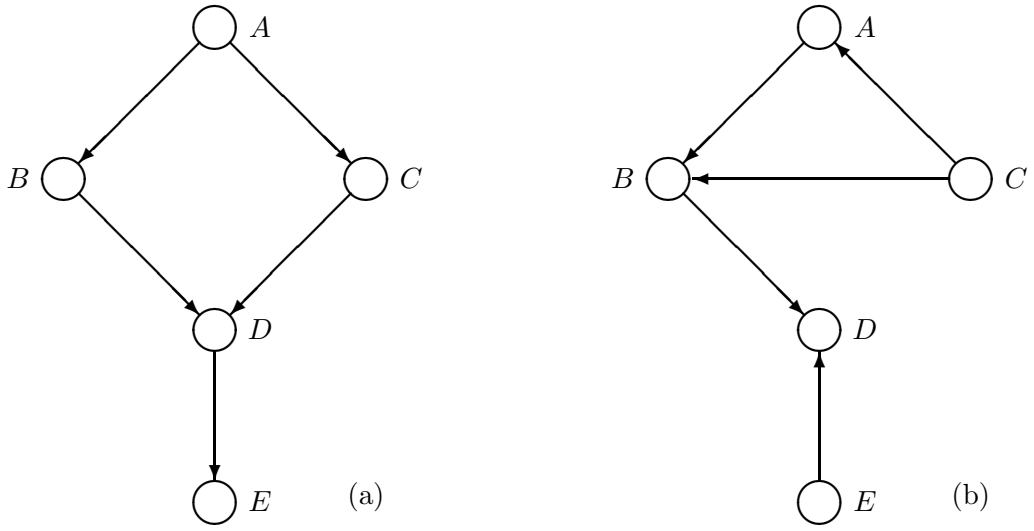


Figure 6.2. According to the structure similarity measure, the distance between the causal graphs (a) and (b) is 6.

	A	B	C	D	E
A	1	0	0	0	0
B	1	1	0	0	0
C	1	0	1	0	0
D	0	1	1	1	0
E	0	0	0	1	1

(a)

	A	B	C	D	E
A	1	0	1	0	0
B	1	1	1	0	0
C	1	0	0	0	0
D	0	1	0	1	1
E	0	0	0	0	1

(b)

Figure 6.3. The ordered binary structure matrices for causal graphs (a) and (b) in Figure 6.2. Each element $a_{ij} = 1$ represents that the column variable V_j directly causes the row variable V_i.

data point can fall between 0 (same structures) and 72 (different on all off-diagonal elements in the ordered structure binary matrices, i.e., $9 \times 9 - 9$) for our two cases.

The independent variable of the evaluation is the set of systems that support in constructing model structures. The dependent variable is the effectiveness of systems measured by the structure similarity measure. I conduct two single factor ANOVA hypothesis tests over the data points with $\alpha = 0.05$. For each subject j and task i, I compute $D(M_g(c_i), M_{Ima}(c_i, s_j))$, $D(M_g(c_i), M_{Eq}(c_i, s_j))$, and $D(M_g(c_i), M_{Ex}(c_i, s_j))$. Let A, B, and C be the random variables of the structure discrepancies between gold standard models and the models constructed by *ImaGeNIe*, *EqGeNIe* and *Excel* respectively. The null hypotheses are $H_0 : \mu_A = \mu_B = \mu_C$ for both Task I and Task II.

Efficiency

I analyze the efficiency of model constructions. The unit of measure here is the time taken to construct a model for a single case by a single subject. I have three systems (*ImaGeNIe*, *EqGeNIe* and *Excel*), 36 subjects, and two tasks. I therefore have $3 \times 36 \times 2$ data points for statistical analysis.

The independent variable of this analysis is the set of systems that support in constructing models. The dependent variable is the efficiency of systems measured by the time used in completing each task. For each subject i and task j, I recorded the time used in constructing models. I conduct two single factor ANOVA hypothesis tests over the data points with $\alpha = 0.05$. Let A, B, and C be the random variables of the time used in constructing models with *ImaGeNIe*, *EqGeNIe* and *Excel* respectively. The null hypotheses are $H_0 : \mu_A = \mu_B = \mu_C$ for both Task I and Task II.

Usefulness

I analyze the subjective account of usefulness collected from the answers in the questionnaire. The unit of measure here is the usefulness of systems for constructing models of a single case rated by a single subject on an eleven-point scale. I have three systems (*ImaGeNIe*, *EqGeNIe* and *Excel*), 36 subjects, and two tasks. I therefore have $3 \times 36 \times 2$ data points for statistical analysis.

The independent variable of this analysis is the set of systems that support constructing models. The dependent variable is the usefulness of systems rated by the subject for each task. I conduct two single factor ANOVA hypothesis tests over the data points with $\alpha = 0.05$. Let A, B, and C be the random variables of the usefulness in constructing models with *ImaGeNIe*, *EqGeNIe* and *Excel* respectively. The null hypotheses are $H_0 : \mu_A = \mu_B = \mu_C$ for both Task I and Task II.

6.2 Results

Effectiveness

I report the analysis results for the effectiveness of using three systems in Table 6.2 and 6.3. We see that $F = 0.184 < F$-critical=3.083 and P-value $= 0.832 > \alpha = 0.05$ for Task I and $F = 1.257 < F$-critical=3.083 and P-value $= 0.289 > \alpha = 0.05$ for Task II. Therefore, we cannot reject both null hypotheses $H_0 : \mu_A = \mu_B = \mu_C$ for Tasks I and II.

I also report the descriptive statistics for the effectiveness of using three systems in Table 6.4 and 6.5.

Table 6.2. ANOVA Test for the Effectiveness of Task I ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.13	2	0.065	0.184	0.832	3.083
Within Groups	36.944	105	0.352			
Total	37.074	107				

Table 6.3. ANOVA Test for the Effectiveness of Task II ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	13.407	2	6.704	1.257	0.289	3.083
Within Groups	559.778	105	5.331			
Total	573.185	107				

Table 6.4. Descriptive statistics for the effectiveness of Task I. The range of mean is between 0 and 72.

	Excel	EqGeNIe	ImaGeNIe
Mean	0.056	0.083	0.139
Standard Error	0.056	0.083	0.198
Median	0	0	0
Mode	0	0	0
Standard Deviation	0.333	0.5	0.833
Sample Variance	0.111	0.25	0.694
Kurtosis	36	36	36
Skewness	6	6	6
Range	2	3	5
Minimum	0	0	0
Maximum	2	3	5
Sum	2	3	5
Count	36	36	36
Largest(1)	2	3	5
Smallest(1)	0	0	0
Confidence Level (95%)	0.113	0.169	0.282
Confidence Interval (95%)	[-0.057, 0.168]	[-0.086, 0.253]	[-0.143,0.421]

Efficiency

I report the analysis result for the completion time of Task I in Table 6.6. We see that $F = 18.814 > F$-critical=3.083 and P-value $= 1.039e-07 < \alpha = 0.05$. We therefore reject the null hypothesis $H_0 : \mu_A = \mu_B = \mu_C$ for Task I.

To give insight into the difference, I show the descriptive statistics for the completion time of

Table 6.5. Descriptive statistics for the effectiveness of Task II. The range of mean is between 0 and 72.

	Excel	EqGeNIe	ImaGeNIe
Mean	1.111	0.5	0.278
Standard Error	0.538	0.348	0.198
Median	0	0	0
Mode	0	0	0
Standard Deviation	3.196	2.09	1.186
Sample Variance	10.216	4.371	1.406
Kurtosis	8.379	28.039	18.407
Skewness	3.012	5.164	4.321
Range	13	12	6
Minimum	0	0	0
Maximum	13	12	6
Sum	40	18	10
Count	36	36	36
Largest(1)	13	12	6
Smallest(1)	0	0	0
Confidence Level (95%)	1.081	0.707	0.401
Confidence Interval (95%)	[0.03, 2.192]	[-0.207, 1.207]	[-0.123,0.679]

Table 6.6. ANOVA Test for the Completion Time of Task I ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	123.574	2	61.787	18.814	1.039E-07	3.083
Within Groups	344.833	105	3.284			
Total	468.407	107				

Task I in Table 6.7 and plot the means and their 95% confidence levels in Figure 6.4. We see that *Excel* outperforms *EqGeNIe* and *ImaGeNIe* since there is no overlap of the range of *Excel* to the ranges of other systems. *ImaGeNIe* is better than *EqGeNIe* but their ranges are overlapping.

I report the analysis result for the completion time of Task II in Table 6.8. In Table 6.8, we see that $F = 25.149 > F$-critical=3.083 and P-value $= 1.191e-09 < \alpha = 0.05$. We therefore reject the null hypothesis $H_0 : \mu_A = \mu_B = \mu_C$ for Task II.

To give insight into the difference, I show the descriptive statistics for the completion time of Task II in Table 6.9 and plot the means and their 95% confidence levels in Figure 6.5. We see that *ImaGeNIe* outperforms *Excel* and *EqGeNIe* since there is no overlap of the range of *ImaGeNIe* with the ranges of the other systems.

Table 6.7. Descriptive statistics for the completion time of Task. The average completion time for *Excel*, *EqGeNIe*, and *ImaGeNIe* are 3.111, 5.639, and 4.972 minutes respectively.

	Excel	EqGeNIe	ImaGeNIe
Mean	3.111	5.639	4.972
Standard Error	0.202	0.256	0.409
Median	3	5.5	4.5
Mode	3	4	4
Standard Deviation	1.214	1.533	2.455
Sample Variance	1.473	2.352	6.028
Kurtosis	-0.867	-0.856	2.064
Skewness	0.081	0.351	1.346
Range	4	6	11
Minimum	1	3	2
Maximum	5	9	13
Sum	112	203	179
Count	36	36	36
Largest(1)	5	9	13
Smallest(1)	1	3	2
Confidence Level (95%)	0.411	0.519	0.831
Confidence Interval (95%)	[2.7, 3.522]	[5.12, 6.158]	[4.141, 5.803]

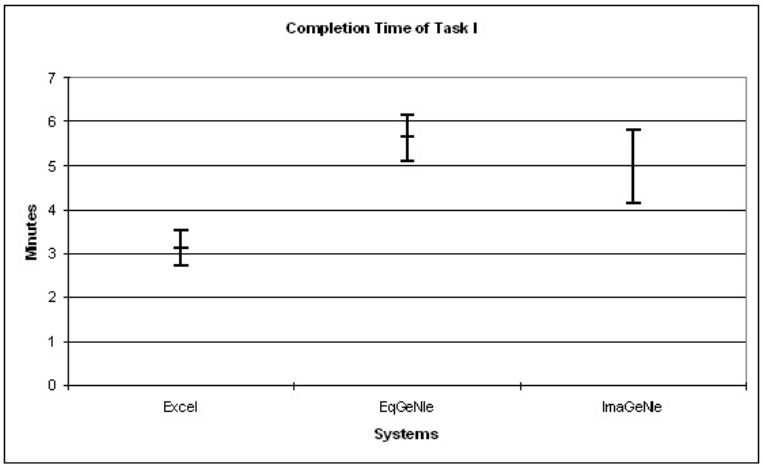


Figure 6.4. Completion time of Task I.

Usefulness

I report the analysis result for the usefulness of the systems for Task I in Table 6.10. We see that $F = 7.393 > F$-critical=3.083 and P-value $= 0.00099 < \alpha = 0.05$. We therefore reject the null hypothesis $H_0 : \mu_A = \mu_B = \mu_C$ for Task I.

To give insight into the difference, I show the descriptive statistics for the usefulness of Task I

Table 6.8. ANOVA Test for the Completion Time of Task II ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	1241.796	2	620.898	25.149	1.191E-09	3.083
Within Groups	2592.305	105	24.688			
Total	3834.102	107				

Table 6.9. Descriptive statistics for the completion time of Task II. The average completion time for *Excel*, *EqGeNIe*, and *ImaGeNIe* are 6.167, 10.389, and 2.083 minutes respectively.

	Excel	EqGeNIe	ImaGeNIe
Mean	6.167	10.389	2.083
Standard Error	0.746	1.18	0.329
Median	5	8.5	1.5
Mode	3	4	1
Standard Deviation	4.475	7.08	1.977
Sample Variance	20.029	50.13	3.907
Kurtosis	3.885	1.008	12.215
Skewness	1.935	1.401	3.239
Range	20	24	10
Minimum	2	4	1
Maximum	22	28	11
Sum	222	374	75
Count	36	36	36
Largest(1)	22	28	11
Smallest(1)	2	4	1
Confidence Level (95%)	1.514	2.396	0.669
Confidence Interval (95%)	[4.653, 7.681]	[7.993, 12.785]	[1.414, 2.752]

Table 6.10. ANOVA Test for the Usefulness of Task I ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	42.667	2	21.333	7.393	0.00099	3.083
Within Groups	303	105	2.886			
Total	345.667	107				

in Table 6.11 and plot the means and their confidence levels in Figure 6.6. We see that *ImaGeNIe* outperforms *EqGeNIe* and *Excel* since there is no overlap of the range of *ImaGeNIe* to the ranges of other systems.

I report the analysis result of the usefulness of the systems for Task II in Table 6.12. In Table 6.12, we see that $F = 24.695 > F$-critical=3.083 and P-value $= 1.621E - 09 < \alpha = 0.05$. I

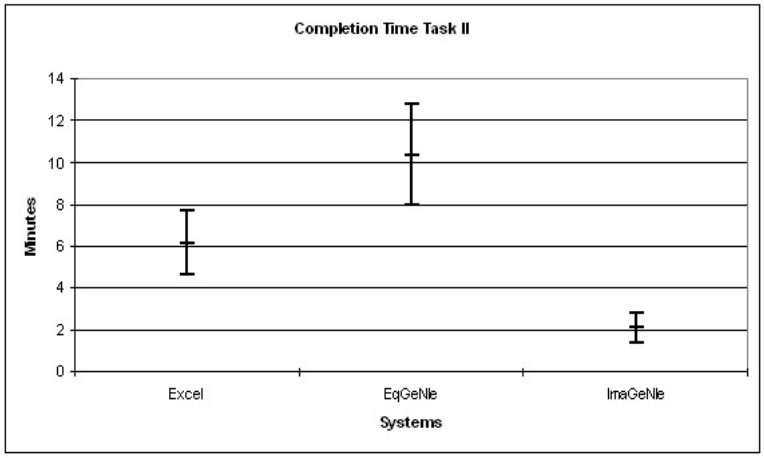


Figure 6.5. Completion time of Task II.

Table 6.11. Descriptive statistics for the usefulness of Task I. The range of mean is between 1 and 10.

	Excel	EqGeNIe	ImaGeNIe
Mean	7.278	7.278	8.611
Standard Error	0.292	0.251	0.304
Median	7	8	9
Mode	8	8	10
Standard Deviation	1.75	1.504	1.825
Sample Variance	3.063	2.263	3.33
Kurtosis	-0.52	1.254	4.45
Skewness	-0.215	-0.984	-1.951
Range	7	7	8
Minimum	3	3	2
Maximum	10	10	10
Sum	262	262	310
Count	36	36	36
Largest(1)	10	10	10
Smallest(1)	3	3	2
Confidence Level (95%)	0.592	0.509	0.617
Confidence Interval (95%)	[6.686, 7.87]	[6.769, 7.787]	[7.994, 9.228]

therefore reject the null hypothesis $H_0 : \mu_A = \mu_B = \mu_C$ for Task II.

To give insight into the difference, I show the descriptive statistics for the usefulness of Task II in Table 6.13 and plot the means and their confidence levels in Figure 6.7. We see that *ImaGeNIe* outperforms *Excel* and *EqGeNIe* since there is no overlap of the range of *ImaGeNIe* to the ranges of other systems.

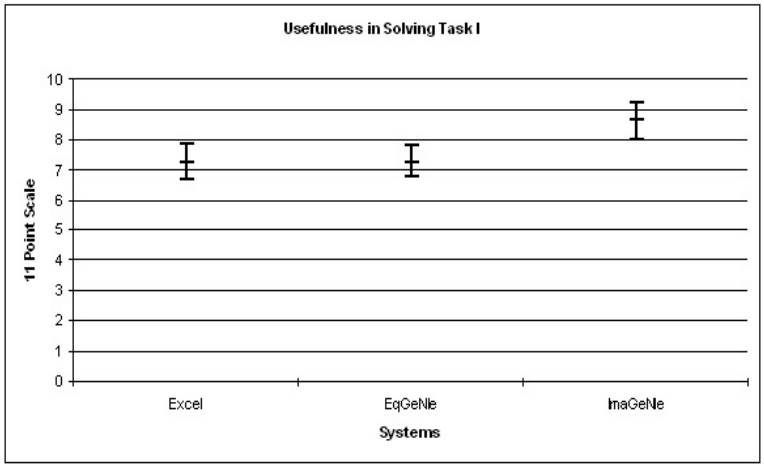


Figure 6.6. Usefulness of Task I.

Table 6.12. ANOVA Test for the Completion Time of Task II ($\alpha = 0.05$).

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	216.352	2	108.176	24.695	1.621E-09	3.083
Within Groups	459.944	105	4.38			
Total	676.296	107				

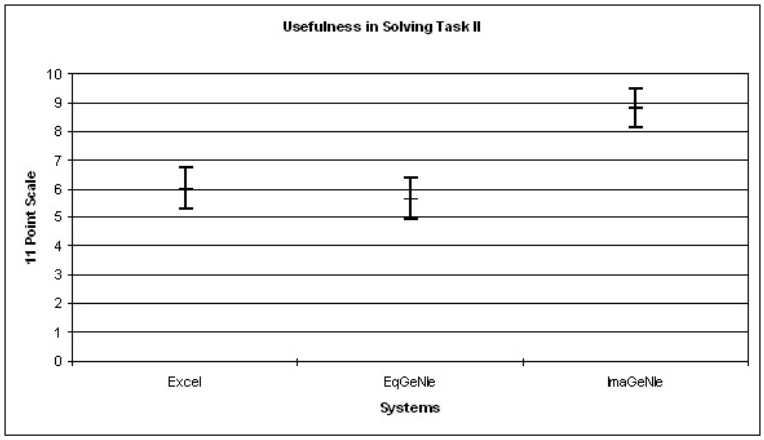


Figure 6.7. Usefulness of Task II.

6.3 Discussion

To summarize my findings, the subjects did find *ImaGeNIe* system an efficient and useful system for casual model construction. Due to the nature of the task, I did not find any significant difference

Table 6.13. Descriptive statistics for the usefulness of Task II. The range of mean is between 1 and 10.

	Excel	EqGeNIe	ImaGeNIe
Mean	6	5.639	8.806
Standard Error	0.359	0.357	0.331
Median	6	5.5	10
Mode	6	8	10
Standard Deviation	2.151	2.14	1.983
Sample Variance	4.629	4.58	3.933
Kurtosis	-0.587	-1.075	4.043
Skewness	-0.255	-0.073	-2.089
Range	8	7	8
Minimum	2	2	2
Maximum	10	9	10
Sum	216	203	317
Count	36	36	36
Largest(1)	10	9	10
Smallest(1)	2	2	2
Confidence Level (95%)	0.728	0.724	0.671
Confidence Interval (95%)	[5.272, 6.728]	[4.915, 6.363]	[8.135, 9.477]

among the effectiveness of model construction using different systems. The reason why *Excel* to outperformed *ImaGeNIe* and *EqGeNIe* might be subjects' familiarity with it. But the fact that *ImaGeNIe* outperformed *Excel* and *EqGeNIe* on Task II suggests that *ImaGeNIe* is helpful in the task of changes in structure.

The first possible threat to this study is the *instrumentation*: the result of the data analysis may be subject to the particular measure over structure discrepancy. So far, I have not found that the structure discrepancy measure is used with structural equations in the literature. I shall reanalyze the data when new measures become available.

The second possible threat to this study is the *experimenter effect*: the designer of *ImaGeNIe* system was the experimenter of this experiment. Although the experimenter has tried to conceal his identity toward subjects and tried to follow the script of the training session rigorously, the experimenter effect might have still possibly sneaked into the training session unconsciously.

Because of the exploratory flavor of this study, I advise readers to consider the number of cases and the number of subjects participated in the study. The answers for the opened question are summarized in Appendix B.5. In general, I had positive responses from users in using *ImaGeNIe*.

Chapter 7
Conclusions

7.1 Summary

Causal models based on structural equations have become a dominant representation for supporting causal reasoning, such as predicting effects of actions, deriving causal relations from data, and generating causal explanations for observed events. Since the quality of recommendations depends not only on the correct use of causal reasoning but also on the quality of models, I focused my work on (1) providing a sound and effective methodology in constructing causal models, (2) supporting the deliberations of the effects of actions with systems containing mixtures of mechanisms, and (3) assisting decision makers in achieving decision objectives by searching for novel actions.

I presented the system *ImaGeNIe* for building graphical causal models based on the extended theory of causal ordering. The mechanism-based view of causality, first proposed by Simon [1953] as the theory of causal ordering, is the theoretical foundation of the implementation of *ImaGeNIe*. Causal ordering explicates the causal relations in a self-contained structure model into a causal graph. I extended the theory of causal ordering to explicate causal relations in an under-constrained structure model such that its graphical representation can represent decision makers' intermediate understanding of a decision problem. The model construction process in *ImaGeNIe* can be viewed as the process of assembling mechanisms from under-constrained models into self-contained models. The models constructed by decision makers using *ImaGeNIe* are guaranteed to be causal because of the mechanism-based view of causality and the decision makers' a-priori assumptions.

In addition to providing decision makers with a sound methodology for building causal models, I assisted decision makers in deliberating effects of actions when one manipulates on systems containing mixtures of mechanisms. I formalized the representations of causal reversibility and the action operator **Act**. I defined the set of effect variables as a property of a mechanism and categorized mechanisms into three categories: completely reversible, partially reversible, and irreversible. And

I further drew an analogy between changes in structure and STRIPS-like action language to define the action operator $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ such that I can assist decision makers in deliberating an action, namely reasoning about which structural equations should be included in $\mathbf{E_{add}}$ or $\mathbf{E_{del}}$. In particular, I developed algorithms to answer two types of queries: (1) When manipulating a causal model, which mechanisms are possibly invalidated and can be removed from the model? (2) Which variables may be manipulated in order to invalidate and, effectively, remove a mechanism from a model?

Although supporting the deliberation over actions is helpful, decision makers still need to provide partial parameters for an action operator, namely $\mathbf{E_{add}}$ or $\mathbf{E_{del}}$, for deliberating an action. I took a step further to address the decision scenarios in which neither $\mathbf{E_{add}}$ nor $\mathbf{E_{del}}$ is given but a causal model and a decision objective. This decision scenario happens when a decision maker, who is confronted with a complex system, does not know which variables to best manipulate or to observe to achieve a desired objective. I refer to this problem as *search for opportunities*, which amounts to both identifying the set of policy variables and computing their optimal setting for a given decision objective. To solve the problem of search for opportunities, I introduced the concept of value of intervention which arises from considering jointly the economic factors and effects of actions in causal models. I proposed augmented causal models, which allow users to specify observability, manipulability, persistence, and focus as properties of variables, to describe a decision problem at hand. I then developed myopic search algorithms to solve the problem of search for opportunities for systems containing mixtures of mechanisms. The algorithm looks one step ahead to compute the value of intervention for each manipulable variable or the value of observation for each observable variables in the model yielding the optimal sequence of actions.

Finally, I have presented the result of an subject experiment evaluating *ImaGeNIe* and found that *ImaGeNIe* can efficiently assist users in constructing causal models for causal reasoning.

Based on these results, I may conclude the central thesis of this dissertation:

> *The mechanism-based view of causality provides an effective formalism for causal model construction and utilization.*

7.2 Future Research

Although structural equations are flexible means of representing mechanisms in causal models and knowledge bases, how to organize structural equations into more higher level knowledge representation is an important research question. Research in Object-Oriented Bayesian Networks (OOBN) [Koller and Pfeffer, 1997; Laskey and Mahoney, 1997; Pfeffer *et al.*, 1999], Multiply-Sectioned Bayesian Networks (MSBN) [Xiang *et al.*, 1993], and Relational Bayesian Networks [Jaeger, 1997], has yielded graphical models based on higher level knowledge representation constructs, but there have been fewer discussions on how to organize these knowledge representation constructs into knowledge bases. The mechanism knowledge base proposed in this dissertation provides the approach of organizing mechanisms hierarchically into subsystems. It is desirable to extend this approach to effectively unify object-hierarchy, type-hierarchy, and entity-relations to derive a more general representation for mechanism knowledge base. Furthermore, providing causal accounts for models constructed with higher level knowledge representation constructs may shed light on the relations between causality different levels of knowledge granularity.

Since the $\mathbf{Act}(\mathbf{E}, \mathbf{E_{pre}}, \mathbf{E_{add}}, \mathbf{E_{del}})$ operator on causal models is drawn from the analogy of the action operator in STRIPS language, I expect that this formalism will open new research directions in the use of causal models and causal reasoning for planning problems. In this dissertation, I proposed the problem of search for opportunities and address the problem by myopic search. The problem of search for opportunities can be considered as a sequential decision problem where the overall utility depends on the sequence of actions. But the problem is not like Markov Decision Process (MDP) or partially observable Markoov Decision Process (POMDP) where a transition model is given [Boutilier *et al.*, 1999]. The problem is also different from decision problems modeled by influence diagrams where the actions are pre-specified and the structures cannot be changed with respect to effects of actions. I believe the formalism of search for opportunities and the **Act** operator have brought a different perspective on planning problems.

Appendix A

Mechanism Knowledge Base Schema

I represent the syntax of mechanism knowledge bases in *ImaGeNIe* using XML Schema.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="mechanism_library">
<xs:complexType> <xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="description" type="xs:string" minOccurs="0"/>
<xs:element ref="subsystem" minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="mechanism" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="version" use="required"/>
<xs:attribute name="id" type="xs:token" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="subsystem">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="description" type="xs:string" minOccurs="0"/>
<xs:element ref="subsystem" minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="mechanism" minOccurs="0" maxOccurs="unbounded"/>
```

```
</xs:sequence>
<xs:attribute name="id" type="xs:token" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="mechanism">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="description" type="xs:string" minOccurs="0"/>
<xs:element name="equation" type="xs:string"/>
<xs:element ref="variable" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:token" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="variable">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="description" type="xs:string" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="id" type="xs:token" use="required"/>
<xs:attribute name="effective" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="observable" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="manipulable" type="xs:boolean" use="optional" default="true"/>
</xs:complexType>
</xs:element>
</xs:schema>
```

Appendix B

Experiment Materials and Data

The following sections present material relevant to the experiments conducted in Chapter 6.

B.1 Stock Market Investment Mechanism Library

Following is the stock marek investment mechanisms library used in the evaluation of *ImaGeNIe* system.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mechanism_library version="1.0" id="StockMechLib">
<name>StockMechLib</name>
<description></description>
<mechanism id="PricePaidMech">
<name>PricePaid</name>
<description></description>
<equation>PP=PPPS*Q+BC</equation>
<variable id="BC" observable="true" manipulable="true" effective="true">
<name>BuyCommission</name>
<description></description>
</variable>
<variable id="PP" observable="true" manipulable="true" effective="true">
<name>PricePaid</name>
<description></description>
</variable>
<variable id="PPPS" observable="true" manipulable="true" effective="true">
```

<name>PurchasePricePerShare</name>

<description></description>

</variable>

<variable id="Q" observable="true" manipulable="true" effective="true">

<name>Quantity</name>

<description></description>

</variable>

</mechanism>

<mechanism id="MarketValueMech">

<name>MarketValue</name>

<description></description>

<equation>MV=MPPS*Q</equation>

<variable id="MPPS" observable="true" manipulable="true" effective="true">

<name>MarketPricePerShare</name>

<description></description>

</variable>

<variable id="MV" observable="true" manipulable="true" effective="true">

<name>MarketValue</name>

<description></description>

</variable>

<variable id="Q" observable="true" manipulable="true" effective="true">

<name>Quantity</name>

<description></description>

</variable>

</mechanism>

<mechanism id="GainMech">

<name>Gain</name>

<description></description>

<equation>G=MV-PP-SC</equation>

<variable id="G" observable="true" manipulable="true" effective="true">

<name>Gain</name>

<description></description>

</variable>

<variable id="MV" observable="true" manipulable="true" effective="true">

<name>MarketValue</name>

<description></description>

</variable>

<variable id="PP" observable="true" manipulable="true" effective="true">

<name>PricePaid</name>

<description></description>

</variable>

<variable id="SC" observable="true" manipulable="true" effective="true">

<name>SaleCommission</name>

<description></description>

</variable>

</mechanism>

<mechanism id="PercentageGainMech">

<name>PercentageGain</name>

<description></description>

<equation>PG=G/PP</equation>

<variable id="G" observable="true" manipulable="true" effective="true">

<name>Gain</name>

<description></description>

</variable>

<variable id="PG" observable="true" manipulable="true" effective="true">

<name>PercentageGain</name>

<description></description>

</variable>

<variable id="PP" observable="true" manipulable="true" effective="true">

<name>PricePaid</name>

```
<description></description>

</variable>

</mechanism>

</mechanism_library>
```

B.2 Task Description

Problem Domain: Stock Market Investment

Variables: The following are variables in our hypothetical stock market:

- PurchasePricePerShare ($PPPS$): the stock price per share when the stock was purchased.
- MarketPricePerShare ($MPPS$): the current stock price per share in stock trading center.
- Quantity (Q): the number of shares.
- BuyCommission (BC): commission paid to brokerage for buying the stock.
- SaleCommision (SC): commission paid to brokerage for selling the stock.
- PricePaid (PP): price paid to buy the stock.
- MarketValue(MV): the stock value according to current market price.
- Gain (G): the gain of realizing the trading.
- PercentageGain (PG): the percentage gain of trading.

Equations: The following are equations that we use to decide our trading strategy:

- $PP = PPPS * Q + BC$
- $MV = MPPS * Q$
- $G = MV - PP - SC$
- $PG = G/PP$

Construct models using *ImaGeNIe* (*EqGeNIe*, *Excel*) to answer the following questions:

Task I Given

- $PPPS = 21.69$
- $MPPS = 23.5$
- $Q = 105$
- $BC = 22.99$
- $SC = 22.99$

  What is your percentage gain?

Task II Assume that you are given the model as in (1). Goal is to make a 20 percentage gain ($PG = 0.2$) on your trading. At what price ($MPPS$) you should sell your stock?

B.3 Training Session

The experimenter shows the following pictures and explains to subjects the physical model of a cart with an object on it:

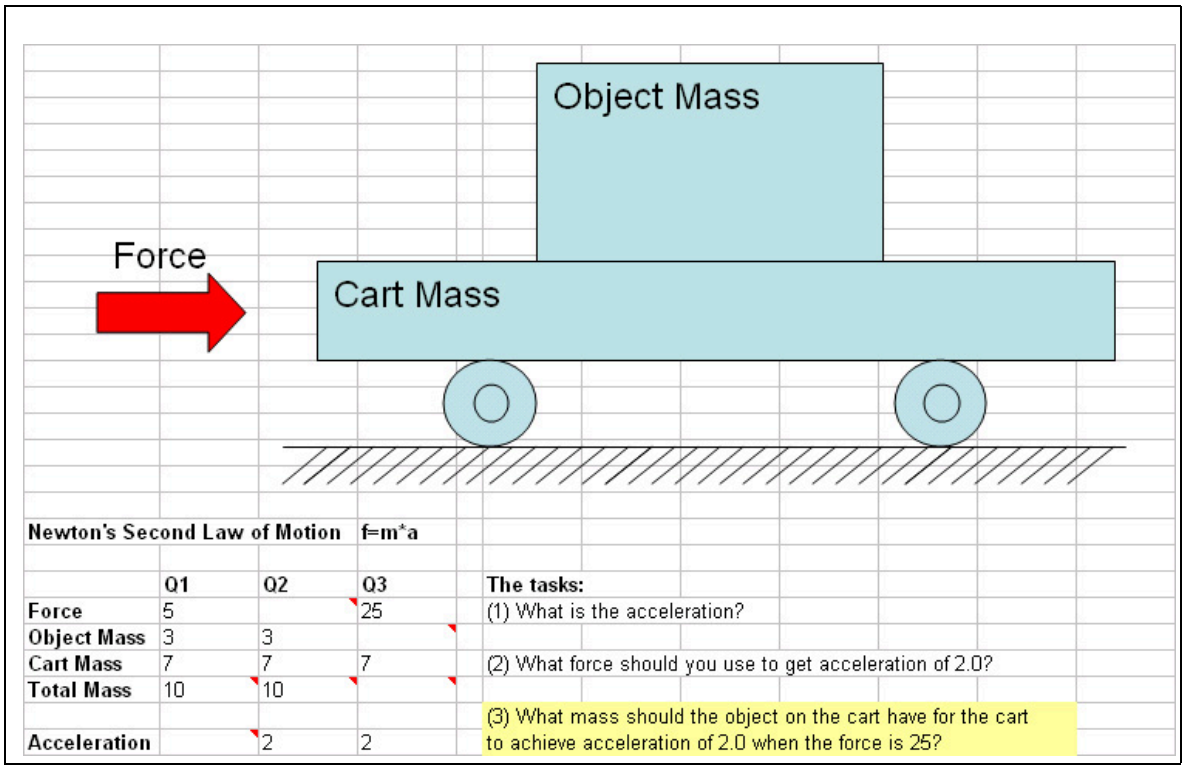


Figure B.1. A training case of Newton's Second Law presented in *Excel*.

- "There is a cart on the table. There is an object on the cart. The total mass is the mass of the cart plus the mass of the object. If we apply force on the cart, the cart will accelerate. The physical law governing this acceleration is Newton's Second Law: $Force = Mass \times Acceleration$, i.e., $f = m \times a$."

Training Session with *Excel*

The experimenter guides subjects to reason the following tasks in *Excel*:

- Task 1: "Assume that the mass of cart is 7 kg and the object of the cart is 3 kg. The total mass of the cart is 10 kg (7 kg + 3 kg). Assume that we apply 5 N. What's the acceleration? The answer is 0.5 m/s^2, which is computed by $Acceleration = Force/TotalMass$."

- Task 2: "Assume that the mass of cart is 7 kg and the object of the cart is 3 kg. The total mass of the cart is 10 kg (7kg + 3kg). What force should we use to get acceleration of 2 m/s^2? The answer is 20 N, which is computed by $Force = TotalMath * Acceleration$."

- Task 3: "Assume that the mass of the cart is 7 kg. What mass should the object on the cart have for the cart to achieve acceleration of 2.0 m/sec^2 when the force is 25 N? The answer is 25 kg, which is computed by $TotalMath = Force/Acceleration$ and $ObjectMath = TotalMath - CartMath$."

Training Session with *EqGeNIe*

The experimenter shows subjects how to use *EqGeNIe* in creating models for the reasoning tasks as follows:

- Select equation node tool. Create nodes *ObjectMass*, *CartMass*, *TotalMass*, *Force*, and *Acceleration* on by one.

- Open node property page for equation nodes, switch to *Definition* page, and change the definition for each task.

- Click on *UpdateBelief* to compute the value for equation nodes and answer the questions in the tasks.

Training Session with *ImaGeNIe*

The experimenter shows subjects how to use *ImaGeNIe* in creating models for the reasoning tasks as follows:

- Apply drag and drop to select mechanism from Mechanism Tree view into *ImaGeNIe* graphical view.

- Use context menu on each mechanism node to control its value.

- View the set of currents equation in equation tool tip of the structure box.

- View the status of the structure box.

- Merge two mechanism nodes from one structure box to another.

- Use context menu on each mechanism node to control its value and release one of the equations if the system is self-contained.
- Use the above techniques to construe the models for each reasoning task.

B.4 Questionnaire

Subject Number: Sex:

1. How do you rate your familiarity in using Microsoft Excel?

extremely unfamiliar ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ extremely familiar

1 5 10

2. How do you rate the usefulness of Microsoft Excel in solving Task 1 and 2?

Please mark 'x' on the circle.

Task 1:

useless ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ extremely useful

1 5 10

Task 2:

useless ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ ◯ extremely useful

1 5 10

3. How do you rate the usefulness of *EqGeNIe* in solving Task 1 and 2?

   Please mark 'x' on the circle.

Task 1:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

Task 2:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

4. How do you rate the usefulness of *ImaGeNIe* in solving Task 1 and 2?

   Please mark 'x' on the circle.

Task 1:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

Task 2:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

5. How do you rate the usefulness of "mechanism libraries" in ImaGeNIe?

   Please mark 'x' on the circle.

Task 1:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

Task 2:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

6. Do you find the graphical view that depicting relations among variables helping you in solving the tasks?

   Please mark 'x' on the circle.

Task 1:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

Task 2:

useless ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ extremely useful

1 5 10

Please rank the ease of use in solving Task I: ($A > B$ denotes that A is easier to use than B).

____ *Excel* > *EqGeNIe* > *ImaGeNIe*

____ *Excel* > *ImaGeNIe* > *EqGeNIe*

____ *EqGeNIe* > *Excel* > *ImaGeNIe*

____ *EqGeNIe* > *ImaGeNIe* > *Excel*

____ *ImaGeNIe* > *Excel* > *EqGeNIe*

____ *ImaGeNIe* > *EqGeNIe* > *Excel*

Please rank the ease of use in solving Task II: ($A > B$ denotes that A is easier to use than B).

____ *Excel* > *EqGeNIe* > *ImaGeNIe*

____ *Excel* > *ImaGeNIe* > *EqGeNIe*

____ *EqGeNIe* > *Excel* > *ImaGeNIe*

____ *EqGeNIe* > *ImaGeNIe* > *Excel*

____ *ImaGeNIe* > *Excel* > *EqGeNIe*

____ *ImaGeNIe* > *EqGeNIe* > *Excel*

Please provide your comments on using different systems in solving each task:

B.5 Comments

Comments compiled from the questionnaire:

Subject 1: [blank]

Subject 2: [blank]

Subject 3: [blank]

Subject 4: EqGeNIe crashed and in order to calculate the second problem, it seems that the entire diagram had to be calculated again which made it not so user friendly. ImaGeNIe was very helpful in solving the second task, after the first was completed. All you had to do is clear one variable and you were finished (very useful).

Subject 5: For Task I, EqGeNIe was slightly easier than Excel because of the typing involved in Excel. ImaGeNIe was the most difficult but not overly so. Familiarity with the 2 GeNIe tools would help in modeling the problem. For task 2, EqGeNIe was by far the easiest with Excel being by far the most difficult. I like EqGeNIe the most in helping to model and solve the problems.

Subject 6: ImaGeNIe is really good and take out a lot of the thinking process making it easier. Equation GeNIe makes you think a lot more. Excel works fine but isn't much fun. I'd rate ImaGeNIe as the best.

Subject 7: ImaGeNIe is very useful, it decreases the mental work load, calculation becomes easy. EqGeNIe is only good when all the controls (variables) are given on the right-hand side of the equation; when I need to re-formulate the equation, modeling just takes some time. It is good to try ImaGeNIe in person, when I saw it at the beginning I was a little bit confused. But after I try it, I found it's very good.

Subject 8: If I am trying to solve a different type of problem. The preference of the tools might be different.

Subject 9: ImaGeNIe is really good very easy to use and takes care of everything by itself.

Subject 10: [blank]

Subject 11: ImaGeNIe seemed pretty straightforward and with some experience one can move around pretty quickly. I like principle behind EqGeNIe and recognize that my existing familiarity with GeNIe might help me a bit. The second problem was a bit confusing at first, until I figure out how to reverse the equations. With practice, this can be pretty useful also. In my obviously limited experiences, the ImaGeNIe was a little more intuitive. I think the only reason that I zipped through Excel so quickly was because it was familiar.

Subject 12: The ImaGeNIe graphical tool to learn dependencies between variable in any field, good learning tool.

Subject 13: Very nice graphical presentation. Very intuitive.

Subject 14: You should ask about background to compare users. I.E. Engineering Computers.

Subject 15: Advantage of Excel is that when I was setting variables I could easily remember what I already set because the organization of rows and columns helped me remember better than freely motioned nodes in EqGeNIe. In case of ImaGeNIe it was not such problem because I did not have to check my setting because I know previous result and it was enough to check result. Changing order of tools might made EqGeNIe better. It took me longer to check variable because of checking values by positioning the mouse over small icons on node picture.

Subject 16: The ImaGeNIe seems to be very useful tool but it is confusing a little bit. Instead of eliminating formulas I was looking for setting correct formula.

Subject 17: ImaGeNIe is good. EqGeNIe is not much useful than Excel when restructuring is required. Its only advantage seems its graphical presentation.

Subject 18: In EqGeNIe, if there are a lot of nodes and complicated relations, it is hard to understand the graphic. Furthermore, I think it would be better if the equation can show in the graphic (not only show when I double click on the node). I often forget what node I set equation and what node I do not.

Subject 19: ImGeNIe is powerful. However, when the number of variables increases, is it still easy to sort everything out and combine them? Preset the mechanism libraries is central to the problem solving. When there are too many equations, it is hard. Better have some categories.

Subject 20: [blank]

Subject 21: [blank]

Subject 22: EqGeNIe is hard to set up. A lot of work for something you could do faster another way. ImaGeNIe made task II much easier, and I thought it was pretty slick. Some questions I might have about it is can you hid pieces of a complex set of equations as to avoid clutter. Also, is there a way to allow interactions between different equations without putting them in the same box?

Subject 24: I did not find the graphical view helpful, it just added another layer of complexity. Doing this task and being unfamiliar with the topic created no benefit to the visualization. The idea was abstract, the relationships were unfamiliar - so it was easier for me to be abstract in my thinking. GeNIe would be better applied when you understand the topic and relationships.

Subject 25: Although ImaGeNIe has some errors, it is relatively easy to use and helps to complete the questions, preventing errors. If the goal is solving the problem, it is good. If the goal is understanding the equation, EqGeNIe is better. Excel is still good since it is easier to see the text of the equations.

Subject 26: Very nice user interface in ImaGeNIe.

Subject 27: ImaGeNIe I feel is very intuitive and easy to learn software. The list of equations on the left makes it simple to keep focus on the task. I was definitely most comfortable with Excel, that is because of the number of years of experiences I have with it. EqGeNIe I found the most hard to work with. I got confused while re-drawing arcs. When I fed a wrong value in EqGeNIe, it simply did not know what to do, there was no error message generated.

Subject 28: I was easily confused using EqGeNIe, especially when rearranging the equations away from their structure shown in the problem domain.

Subject 29: I think the only reason Excel is rated higher than EqGeNIe is base on the familiarity. I know GeNIe well enough, but I have used Excel so often.

Subject 30: EqGeNIe was extremely difficult when restructuring the equation. It is frustrating that the equation is rested when the links are deleted. It was however very easy to set up. ImaGeNIe was opposite that it was not easy to set up but extremely easy to restructure.

Subject 31: I think the hardest part of solving the problems with GeNIe was actually keeping tack of which variables where which, because the arrows crossed so much. Other than figuring out where to put the nodes, the GeNIe tools made the problem easier.

Subject 32: Good when it is inference more complicate problem. But the process of constructing is not that convenient. You should forbid two different name node converge.

Subject 33: ImaGeNIe is good (easy to use), but EqGeNIe is bad (not easy to use).

Subject 34: ImaGeNIe is pretty helpful when we have relationships in hand and we can flexibly find any variables in those equations easily. I like it!! : Good job.

Subject 35: For learning about how the networks work ImaGeNIe is good, because it creates the arcs automatically. For task 2, it is definitely the best, although I did not at first get what was going on in the BG.

Subject 36: I think that both EqGeNIe and ImaGeNIe are very useful tools and not very difficult to use. They were both easy to learn the basics of each. I liked being able to model the equation and being able to manipulate the models in order to solve for different variables.

Subject 37: ImaGeNIe was straight forward to use after one example. It is more intuitive and user friendly than Excel. The drag and drop functionality of ImaGeNIe is great. This capability reduces the amount of steps the user takes in solving the problem. With fewer steps, there is less chance that the user will use the incorrect formula. This is a distnic possibility in the other software used in the test. During the test I only noticed one thing that would be useful,

the addition of an "undo" button. This would undo the last step entered. This (ImaGeNIe) is an impressive tool. It made the task completion easy to perform and visualize.

Subject 38: I found the ImaGeNIe hard to do. I needed to back out and start over again to complete the task. With training it might be better but not really for short term learning.

Bibliography

[Balke and Pearl, 1995] Alexander Balke and Judea Pearl. Counterfactuals and policy analysis in structural models. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pages 11–18, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[Bogers, 1997] Jeroen J.J. Bogers. Supporting the change in structure in a decision support system based on structural equations. Master's thesis, Department of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, August 1997.

[Boutilier *et al.*, 1999] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[Breese and Heckerman, 1996] John S. Breese and David Heckerman. Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 124–132, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[Breese *et al.*, 1994] John S. Breese, Robert P. Goldman, and Michael P. Wellman. Introduction to the special section on knowledge-based construction of probabilistic and decision models. *IEEE Transactions on Systems, Man and Cybernetics*, 24(11):1577–1579, 1994.

[Cooper and Herskovits, 1991] Gregory F. Cooper and Edward Herskovits. A Bayesian method for constructing Bayesian belief networks from databases. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–91)*, pages 86–94, San Mateo, California, 1991. Morgan Kaufmann Publishers.

[Cooper and Herskovits, 1992] Gregory F. Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[Cooper, 1999] Gregory F. Cooper. An overview of the representation and discovery of causal relationships using Bayesian networks. In Glymour and Cooper [1999], chapter one, pages 3–62.

[Cooper, 2000] Gregory Cooper. A Bayesian method for causal modeling and discovery under selection. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 98–106, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[Dash and Druzdzel, 2001] Denver Dash and Marek J. Druzdzel. Caveats for causal reasoning with equilibrium models. In Salem Benferhat and Philippe Besnard, editors, *Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2001)*, volume 2143 of *Lecture Notes in Artificial Intelligence*, pages 192–203, Toulouse, France, 2001. Springer-Verlag.

[de Kleer and Brown, 1986] Johan de Kleer and John Seely Brown. Theories of causal ordering. *Artificial Intelligence*, 29:33–61, 1986.

[Diez, 1993] Francisco Javier Diez. Parameter adjustment in Bayes networks. The generalized noisy OR-gate. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 99–105, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[Druzdzel and Díez, 2003] Marek J. Druzdzel and F. Javier Díez. Combining knowledge from different sources in causal probabilistic models. *JMLR Special Issue on the Fusion of Domain Knowledge with Data for Decision Support*, 4:295–316, July 2003.

[Druzdzel and Simon, 1993] Marek J. Druzdzel and Herbert A. Simon. Causality in Bayesian belief networks. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 3–11, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[Druzdzel and van der Gaag, 2000] Marek J. Druzdzel and Linda C. van der Gaag. Building probabilistic networks: "Where do the numbers come from?" guest editors' introduction. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):481–486, July–August 2000.

[Druzdzel and van Leijen, 2001] Marek J. Druzdzel and Hans van Leijen. Causal reversibility in Bayesian networks. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(1):45–62, Jan 2001.

[Druzdzel, 1992] Marek J. Druzdzel. *Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense*. PhD thesis, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, December 1992.

[Even, 1979] Shimon Even. *Graph algorithms*. Computer Science Press, Potomac, MD, 1979.

[Fikes and Nilsson, 1971] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.

[Galles and Pearl, 1997] D. Galles and J. Pearl. Axioms of causal relevance. *Artificial Intelligence*, 97(1–2):9–43, 1997.

[Glymour and Cooper, 1999] Clark Glymour and Gregory F. Cooper, editors. *Computation, Causation, and Discovery*. AAAI Press, Menlo Park, CA, 1999.

[Goldberger, 1972] Arthur S. Goldberger. Structural equation methods in the social sciences. *Econometrica*, 40(6):979–1001, November 1972.

[Haavelmo, 1943] Trygve Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11(1):1–12, January 1943.

[Haavelmo, 1944] Trygve Haavelmo. The probability approach in econometrics. *Econometrica*, 12(Issue Supplement):1–115, July 1944.

[Halpern and Pearl, 2001] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach — part 1: Causes. In *Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference (UAI-2001)*, pages 194–202, San Francisco, CA, 2001. Morgan Kaufmann Publishers.

[Heckerman and Shachter, 1995] David Heckerman and R. Shachter. Decision-theoretic foundations for causal reasoning. *Journal of Artificial Intelligence Research*, 3:405–430, 1995.

[Heckerman, 1990] David Heckerman. Probabilistic similarity networks. *Networks*, 20(5):607–636, August 1990.

[Henrion, 1989] Max Henrion. Some practical issues in constructing belief networks. In L.N. Kanal, T.S. Levitt, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 3*, pages 161–173. Elsevier Science Publishing Company, Inc., New York, N. Y., 1989.

[Hood and Koopmans, 1953] William C. Hood and Tjalling C. Koopmans, editors. *Studies in Econometric Method. Cowles Commission for Research in Economics. Monograph No. 14.* John Wiley & Sons, Inc., New York, NY, 1953.

[Howard and Matheson, 1981] Ronald A. Howard and James E. Matheson. Influence diagrams. In Ronald A. Howard and James E. Matheson, editors, *The Principles and Applications of Decision Analysis*, pages 719–762. Strategic Decisions Group, Menlo Park, CA, 1981.

[Howard, 1966] Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1):22–26, August 1966.

[Howard, 1971] Ronald A. Howard. Proximal decision analysis. *Management Science*, 17(9):505–541, May 1971.

[Iwasaki and Simon, 1986a] Yumi Iwasaki and Herbert A. Simon. Causality in device behavior. *Artificial Intelligence*, 29(1):3–32, July 1986.

[Iwasaki and Simon, 1986b] Yumi Iwasaki and Herbert A. Simon. Theories of causal ordering: Reply to de Kleer and Brown. *Artificial Intelligence*, 29(1):63–67, July 1986.

[Iwasaki and Simon, 1994] Yumi Iwasaki and Herbert A. Simon. Causality and model abstraction. *Artificial Intelligence*, 67(1):143–194, May 1994.

[Iwasaki, 1988] Yumi Iwasaki. *Model Based Reasoning of Device Behavior With Causal Ordering.* PhD thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, August 1988.

[Jaeger, 1997] Manfred Jaeger. Relational Bayesian networks. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 266–273, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[Jensen and Vomlelová, 2002] Finn V. Jensen and Marta Vomlelová. Unconstrained influence diagrams. In *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–2002)*, pages 234–241, San Francisco, CA, 2002. Morgan Kaufmann Publishers.

[Keeney, 1994] Ralph L. Keeney. *Value-focused Thinking: a Path to Creative Decision Making.* Harvard University Press, Cambridge, MA, 1994.

[Koller and Pfeffer, 1997] Daphne Koller and Avi Pfeffer. Object-oriented Bayesian networks. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 302–313, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[Koopmans, 1950] Tjalling C. Koopmans, editor. *Statistical Inference in Dynamic Economic Models. Cowles Commission for Research in Economics. Monograph No. 10.* John Wiley & Sons, Inc., New York, NY, 1950.

[Laskey and Mahoney, 1997] Kathryn Blackmond Laskey and Suzanne M. Mahoney. Network fragments: Representing knowledge for constructing probabilistic models. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pages 334–341, San Francisco, CA, 1997. Morgan Kaufmann Publishers.

[Mahoney and Laskey, 1996] Suzanne M. Mahoney and Kathryn B. Laskey. Network engineering for complex belief networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 389–396, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[Marschak, 1953] Jacob Marschak. Economic measurements for policy and prediction. In Hood and Koopmans [1953], chapter I, pages 1–26.

[Matheson, 1990] James E. Matheson. Using influence diagrams to value information and control. In *Influence Diagrams, Belief Nets and Decision Analysis*, chapter 2, pages 25–48. John Wiley & Sons Ltd., 1990.

[Murphy, 2001] Kevin Murphy. Active learning of causal Bayes net structure. Technical report, U.C. Berkeley, March 2001.

[Nayak, 1992] P. Pandurang Nayak. *Automated Modeling of Physical Systems*. Dissertation, Department of Computer Science, Stanford University, Stanford, California, September 1992.

[Nayak, 1994] P. Pandurang Nayak. Causal approximations. *Artificial Intelligence*, 70(1–2):1–58, 1994.

[Papadimitriou and Stelglitz, 1982] Christos H. Papadimitriou and Kenneth Stelglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, Inc., 1982.

[Pearl and Robins, 1995] Judea Pearl and James Robins. Probabilistic evaluation of sequential plans from causal models with hidden variables. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pages 444–453, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[Pearl and Verma, 1991] Judea Pearl and Thomas S. Verma. A theory of inferred causation. In J.A. Allen, R. Fikes, and E. Sandewall, editors, *KR–91, Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452, Cambridge, MA, 1991. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[Pearl, 1993] Judea Pearl. Comment: Graphical models, causality, and intervention. *Statistical Science*, 8:266–269, 1993.

[Pearl, 1994] Judea Pearl. A probabilistic calculus of actions. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, pages 454–462, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[Pearl, 1995] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–710, 1995.

[Pearl, 1998] Judea Pearl. Graphs, causality, and structural equation models. *Socioligical Methods and Research*, 27(2):226–284, Nov 1998.

[Pearl, 2000] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.

[Pfeffer *et al.*, 1999] Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa. SPOOK: A system for probabilistic object-oriented knowledge representation. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99)*, pages 541–550, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[Phillips, 1984] L. D. Phillips. A theory of requisite decision models. *Acta Psychologica*, 56:29–48, 1984.

[Pradhan *et al.*, 1996] Malcolm Pradhan, Max Henrion, Gregory Provan, Brendan Del Favero, and Kurt Huang. The sensitivity of belief networks to imprecise probabilities: An experimental investigation. *Artificial Intelligence*, 85(1-2):363–397, 1996.

[Russell and Norvig, 1995] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[Savage, 1972] Leonard J. Savage. *The Foundations of Statistics (Second Revised Edition)*. Dover Publications, New York, NY, 1972.

[Serrano and Gossard, 1987] D. Serrano and D. Gossard. Constraint management in conceptual design. In D. Sriram and R.A. Adey, editors, *Knowledge Based Expert Systems in Engineering: Planning and Design*, pages 211–224. Computational Mechanics Publications, Southampton, 1987.

[Simon and Iwasaki, 1988] Herbert A. Simon and Yumi Iwasaki. Causal ordering, comparative statics, and near decomposability. *Journal of Econometrics*, 39(1–2):149–173, SEP/OCT 1988.

[Simon and Rescher, 1966] Herbert A. Simon and Nicholas Rescher. Cause and counterfactual. *Philosophy of Science*, 33(4):323–340, December 1966.

[Simon *et al.*, 2000] Herbert A. Simon, Jayant R. Kalagnanam, and Marek J. Druzdzel. Performance budget planning: The case of a research university. In preparation, 2000.

[Simon, 1953] Herbert A. Simon. Causal ordering and identifiability. In Hood and Koopmans [1953], chapter III, pages 49–74.

[Simon, 1979] Herbert A. Simon. The meaning of causal ordering. In Robert K. Merton, James S. Coleman, and Peter H. Rossi, editors, *Qualitative and Quantitative Social Research: Papers in Honor of Paul F. Lazarsfeld*, chapter 8, pages 65–81. The Free Press, A Division of Macmillan Publishing Co., Inc., 1979.

[Spirtes *et al.*, 1993] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Springer Verlag, New York, 1993.

[Spirtes *et al.*, 1995] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pages 499–506, San Francisco, CA, 1995. Morgan Kaufmann Publishers.

[Spirtes *et al.*, 2000] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA, second edition, 2000.

[Srinivas, 1993] Sampath Srinivas. A generalization of the noisy-OR model. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*, pages 208–215, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

[Strotz and Wold, 1960] Robert H. Strotz and H.O.A. Wold. Recursive vs. nonrecursive systems: An attempt at synthesis; Part I of a triptych on causal chain systems. *Econometrica*, 28(2):417–427, April 1960.

[Suermondt, 1992] Henri J. Suermondt. *Explanation in Bayesian Belief Networks*. PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA, March 1992.

[Tong and Koller, 2001] Simon Tong and Daphne Koller. Active learning for structure in Bayesian networks. In *International Joint Conference on Artificial Intelligence*, pages 863–869, Seattle, Washington, August 2001.

[von Winterfeldt and Edwards, 1988] Detlof von Winterfeldt and Ward Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, 1988.

[Wold and Jureen, 1953] Herman Wold and Lars Jureen. *Demand Analysis. A Study in Econometrics*. JohnWiley and Sons, Inc., New York, 1953.

[Wold, 1954] Herman Wold. Causality and econometrics. *Econometrica*, 22(2):162–177, April 1954.

[Wold, 1964] H.O.A. Wold. On the definition and meaning of causal concepts. In H.O.A. Wold, editor, *Model Building in the Human Sciences*, pages 265–295. Union Europeenne Deditions, Monaco, 1964.

[Wright, 1934] Sewall Wright. The method of path coefficients. *Annals of Mathematical Statistics*, 5:161–215, 1934.

[Xiang *et al.*, 1993] Yang Xiang, David Poole, and Michael P. Beddoes. Multiply sectioned Bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence*, 9(2):171–220, 1993.

[Yoo and Cooper, 2002] Changwon Yoo and Gregory F. Cooper. Discovery of gene-regulation pathways using local causal search. In *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*, pages 914–918, 2002.