



# Interactive construction of graphical decision models based on causal mechanisms

Tsai-Ching Lu<sup>a,\*</sup>, Marek J. Druzdzel<sup>a,b</sup>

<sup>a</sup> Decision Systems Laboratory, School of Information Sciences and Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA

<sup>b</sup> Faculty of Computer Science, Białystok Technical University, Wiejska 45A, 15-351 Białystok, Poland

## ARTICLE INFO

### Article history:

Received 30 November 2007

Accepted 31 January 2009

Available online 29 March 2009

### Keywords:

Decision analysis

Model building

Bayesian networks

Structural equation models

Causal ordering

## ABSTRACT

We propose a framework for building graphical decision models from individual causal mechanisms. Our approach is based on the work of Simon [Simon, H.A., 1953. Causal ordering and identifiability. In: Hood, W.C., Koopmans, T.C. (Eds.), *Studies in Econometric Method*. Cowles Commission for Research in Economics. Monograph No. 14. John Wiley and Sons Inc., New York, NY, pp. 49–74 (Ch. III)], who proposed a causal ordering algorithm for explicating causal asymmetries among variables in a self-contained set of structural equations. We extend the causal ordering algorithm to under-constrained sets of structural equations, common during the process of problem structuring. We demonstrate that the causal ordering explicated by our extension is an intermediate representation of a modeler's understanding of a problem and that the process of model construction consists of assembling mechanisms into self-contained causal models. We describe IMAgeNIE, an interactive modeling tool that supports mechanism-based model construction and demonstrate empirically that it can effectively assist users in constructing graphical decision models.

© 2009 Elsevier B.V. All rights reserved.

## 1. Introduction

Probabilistic Graphical Models, such as influence diagrams (Howard and Matheson, 1981) and Bayesian networks (Pearl, 1988), have become popular modeling tools for supporting decision making under uncertainty. Their ability to efficiently represent joint probability distributions over tens, hundreds or even thousands of variables makes it possible to build decision models involving systems of challenging complexity, hard to imagine with modeling tools such as decision trees. Since the normative character of graphical decision models guarantees the correctness of inference procedures, the quality of advice derived from such models directly depends on the quality of the models. 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). As the notion of requisiteness is fairly subjective, building requisite models requires a good measure of human intuition and creativity. Effectively, construction of graphical decision models is demanding in terms of human expertise. This calls for sound methodologies and effective tools to facilitate problem structuring and parameter elicitation. Although support for obtaining model parameters, such as probability distribution and utilities, has received some attention in behavioral decision theory literature (e.g., von Winterfeldt and Edwards, 1986) and in artificial intelligence (e.g., Druzdzel and van der Gaag, 2000), there are some indi-

cations that the quality of advice is more sensitive to the model structure than to the precision of its numerical parameters (Druzdzel and Oniško, 2008a,b; Oniško and Druzdzel, 2003; Pradhan et al., 1996).

Prior efforts in aiding model structuring in artificial intelligence can be classified into four approaches. The *first* approach focuses on *more expressive structure representations*. Instances of the Independence of Causal Influences (ICI) gates, such as the Noisy-OR (Pearl, 1988; Henrion, 1989) and its generalization Noisy-MAX (Diez, 1993; Maaskant and Druzdzel, 2008; Srinivas, 1993), simplify elicitation of numerical probabilities, so do *similarity network* and *partitioning* for representing *subset independence* (Heckerman, 1990). Object-oriented Bayesian networks (OOBN) (Koller and Pfeffer, 1997; Pfeffer et al., 1999) provide a way of reusing fragments of models across problem instances. Other approaches (Poole, 2003; Laskey, 2004) draw on intuitive foundations of first-order logic. The *second* approach, usually referred to as *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 (Holtzman, 1989; 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). The *fourth* approach, most related to our work, is to apply *system and knowledge engineering* techniques for aiding the process of building graphical models. Mahoney and Laskey (1996) address the issues of modularization, object-oriented design, knowledge-base, and evaluation in a spiral model development cycle.

\* Corresponding author. Present address: HRL Laboratories, LLC, Malibu, CA 90265, USA.

E-mail addresses: [tlu@hrl.com](mailto:tlu@hrl.com) (T.-C. Lu), [marek@sis.pitt.edu](mailto:marek@sis.pitt.edu) (M.J. Druzdzel).

In decision analysis, Mingers and Rosenhead (2004) summarize various problem structuring methods and their applications. Two approaches closely related to ours are causal maps (see Montibeller and Belton (2006) for a review) and influence diagrams (Howard and Matheson, 1981). Causal maps are directed graphs that reflect decision-makers' thinking of means and ends of their decision problems. Nadkarni and Shenoy (2004) demonstrate an approach to convert causal maps into Bayesian networks by checking conditional independence and non-circular relations among concepts. Howard and Matheson (1981) emphasize decision analysis cycle where a problem is represented by decision alternatives, information uncertainties, and preferences of the decision maker. Both causal maps and influence diagrams employ the notion of causal influences in their model structuring, i.e., decision makers suppose to reason upon their problem in causal terms, normally facilitated or guided by knowledge engineers. Such exploratory modeling process can be further assisted by more rigorous methodologies, such as value focused thinking (Keeney, 1992). However, it quickly becomes a daunting task for decision makers to reason upon tens or hundreds variables and to verify their direct and indirect causal relations.

In our framework, we take the mechanism-based view of causality (Simon, 1953; Druzdzel, 1992) where causality is defined within models and causal asymmetries arise when mechanisms are placed in the context of a system. The building blocks of a model are individual causal mechanisms represented by *structural equations* (as shown in Druzdzel and Simon (1993), families of nodes in Bayesian networks can be viewed as encodings of structural equations). Our framework supports encoding mechanisms as functional relations and, wherever causal mechanisms are asymmetric, the direction of causal influence among variables. As causal mechanisms encode our understanding of local interactions among variables, often based on the laws of physics, economics, etc., they are fairly model independent and can be reused in various models. Mechanisms can be organized hierarchically (Iwasaki and Simon, 1994), similar to the abstraction of object-hierarchy, so that domain knowledge can be compiled as libraries of mechanisms. Effectively, mechanism-based view of causality provides a valuable heuristic for acquiring and managing domain knowledge, i.e., mechanisms can be contemplated even without particular decision problems.

In the process of model structuring, when not all mechanisms have been brought into the picture, we are typically dealing with under-constrained systems, i.e., systems of equations that cannot yet be solved because they contain fewer equations than variables. The main theoretical contribution of this paper is extending Simon's causal ordering algorithm to under-constrained systems. Our *extended causal ordering algorithm* explicates causal relations in under-constrained systems so that our framework can assist knowledge engineer and decision maker in exploring, aggregating, and manipulating mechanisms in order to make a model self-contained.

There are two advantages of our framework, compared to existing methods. Firstly, the resulting model structures generated by our framework are guaranteed to be causal, as defined by mechanism-based view of causality, if the underlying structural equations reflect causal mechanisms of the modeled problem. Secondly, the framework functions as a decision facilitator by providing meaningful suggestions on how to expand the model and how to make it self-contained. Our framework assists decision makers in (1) identifying a set of mechanisms related to the current model and bringing them into model workspace, (2) aggregating the newly added mechanisms with the model under construction, (3) specifying the exogenous variables, and (4) extracting reusable mechanisms from existing models into the libraries of mechanisms. We implemented the proposed framework

in an interactive modeling tool that we call IMAgeNIE. We conducted an empirical evaluation of IMAgeNIE and found that it can effectively assist users in constructing causal models.

## 2. Causal ordering

We consider *models* as abstraction of *systems*, pieces of the real world that can be reasonably studied in isolation. A model can be represented by a set of structural equations,  $\mathbf{E} = \{e_1, e_2, \dots, e_m\}$ , where each *structural equation*,  $e_i(V_1, V_2, \dots, V_n) = 0$ , describes a conceptually distinct *mechanism* active in the system. Such models are known as Simultaneous Equation Models (Simon, 1953), or Structural Equation Models (Goldberger, 1972) in economics and social sciences. We normally postulate mechanisms with *stability* criterion to screen off *transitory* relations and assert *autonomous* assumption 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, p. 26).

An equation may be solvable for any of its variables in the algebraic sense. However, causal interpretation of an explicit form of a structure equation requires more care. Simon (1979), as well as others (Haavelmo, 1944; Wold, 1954), believe that different a priori assumptions 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 under different a priori assumptions. He suggested three sources in asserting asymmetries for causal mechanisms: *manipulability*, *time precedence*, and *prepotency*.<sup>1</sup> In other words, an explicit function  $V_i = f_i(V_1, \dots, V_{i-1}, V_{i+1}, \dots, V_n)$  for a structural equation  $e$  represents a causal mechanism where the set of variables  $\{V_1, \dots, V_{i-1}, V_{i+1}, \dots, V_n\}$  are the causes of the effect variable  $V_i$ .

To ensure that  $\mathbf{E}$  is *consistent* (the solution set of  $\mathbf{E}$  is not empty) and *independent* (none of the equations can be derived algebraically from the others), Simon and Rescher (1966) defined the concept of *structure*:

**Definition 1** (*structure*). A structure is a set of equations  $\mathbf{E}$  where  $|\mathbf{E}| \leq |\mathbb{V}(\mathbf{E})|$  such that in any subset  $\mathbf{E}' \subseteq \mathbf{E}$ :

- (1)  $|\mathbf{E}'| \leq |\mathbb{V}(\mathbf{E}')|$ , and
- (2) if the values of any  $|\mathbb{V}(\mathbf{E}')| - |\mathbf{E}'|$  variables in  $\mathbb{V}(\mathbf{E}')$  are chosen arbitrarily, then the values of the remaining  $|\mathbf{E}'|$  variables are determined uniquely, where  $\mathbb{V}(e)$  denotes the set of variables appearing in a structural equation  $e$  and  $\mathbb{V}(\mathbf{E}) = \bigcup_{e \in \mathbf{E}} \mathbb{V}(e)$ .

Simon (1953) proposed an algorithm for explicating asymmetries among variables in a self-contained simultaneous structure equations and producing a causal ordering among variables (equations) in a *self-contained* structure,  $|\mathbf{E}| = |\mathbb{V}(\mathbf{E})|$ . Causal ordering only requires qualitative knowledge of which variables appears in which structural equations, i.e., structure matrix.

**Definition 2** (*Structure matrix* (Simon and Rescher, 1966; Druzdzel and Simon, 1993)). A structure matrix is a qualitative representation of a set of structural equations  $\mathbf{E}$ , where an element  $a_{ij} = x$  if  $V_j \in \mathbf{V}$  participates in  $e_i \in \mathbf{E}$ ; and  $a_{ij} = 0$ , otherwise.

The causal ordering algorithm (COA) takes a self-contained structure  $\mathbf{E}$  as input and outputs a *causal graph*  $G(\mathbf{E}) = (\mathbf{N}, \mathbf{A})$ , where  $\mathbf{N}$  represents variables  $\mathbf{V} = \mathbb{V}(\mathbf{E})$  and  $\mathbf{A}$  is a set of directed

<sup>1</sup> In the later section, we will see that the manipulability plays an important role in our approach of causal modeling.

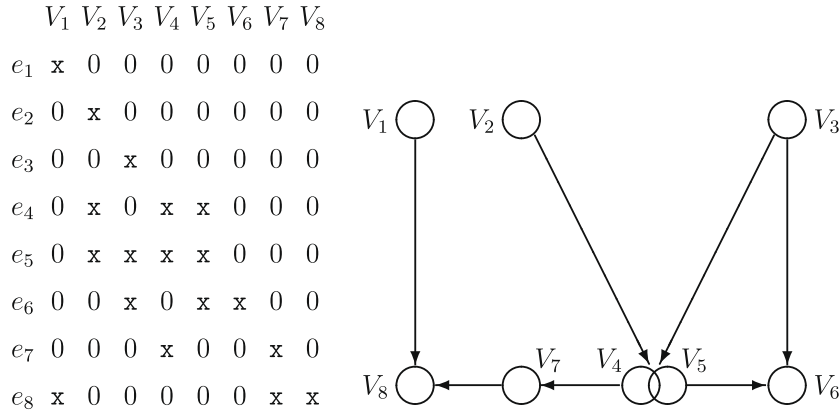


Fig. 1. Causal ordering algorithm takes a self-contained structure as input (left) and outputs a causal graph (right) for Example 1.

arcs among  $\mathbf{N}$ . More precisely,  $\mathbf{N}$  is a partitioning of  $\mathbf{V}$ , meaning that  $\mathbf{N} = \{\mathbf{N}_1, \mathbf{N}_2, \dots, \mathbf{N}_r\}$  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  $V_j \notin \mathbf{N}_i$ . A self-contained structure  $\mathbf{E}$  is *minimal* if it does not contain any proper self-contained subsets. The algorithm starts with *identifying* the minimal self-contained structures in  $\mathbf{E}$ . These identified minimal self-contained structures,  $\mathbf{C}^0 = \{\mathbf{C}_1^0, \mathbf{C}_2^0, \dots, \mathbf{C}_l^0\}$ , are called *complete structures of 0th order* and a partition  $\mathbf{N}_k^0$  on  $\mathbf{V}$  is created for  $\mathbf{V}(\mathbf{C}_k^0)$ , 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 the equations in  $\mathbf{C}_k^0$  need to be solved simultaneously. Next, the algorithm *removes*  $\mathbf{C}^0$  from  $\mathbf{E}$  because the values of  $\mathbf{V}(\mathbf{C}^0)$  are *solved*. We denote the new structure  $\mathbf{E} \setminus \mathbf{C}^0$  as  $\mathbf{E}^1$ . The algorithm then removes the columns representing  $\mathbf{V}(\mathbf{C}^0)$  as *substituting* the solved values of  $\mathbf{V}(\mathbf{C}^0)$  into  $\mathbf{E}^1$  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  $\mathbf{V}(\widehat{\mathbf{C}}_k^p) \setminus \mathbf{V}(\mathbf{C}_k^p)$  to the nodes representing  $\mathbf{N}_k^p$ . Effectively, 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.

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 3** (*Endogenous and exogenous variables*). 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  $V_i \in \mathbf{V}(\widehat{\mathbf{C}}_k^p)$ :

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

The sets of endogenous and exogenous variables pertinent to  $\widehat{\mathbf{C}}_k^p$  are denoted as  $\mathbf{EnV}(\widehat{\mathbf{C}}_k^p)$  and  $\mathbf{ExV}(\widehat{\mathbf{C}}_k^p)$ , respectively.

Simon (1953) uses the concept of endogenous and exogenous variables pertinent to  $\widehat{\mathbf{C}}_k^p$  to define *direct causes*.

**Definition 4** (*direct cause*). For every  $\widehat{\mathbf{C}}_k^p$  in a self-contained structure  $\mathbf{E}$ , each  $V_i \in \mathbf{ExV}(\widehat{\mathbf{C}}_k^p)$  is a direct cause of each  $V_j \in \mathbf{EnV}(\widehat{\mathbf{C}}_k^p)$ .

**Example 1.** Consider the structure matrix presented in Fig. 1. 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 between 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 direct causes of each variable from the set of its parent nodes, e.g.,  $\{V_4, V_5\}$  is directly caused by  $V_2$  and  $V_3$  which directly corresponds to  $\mathbf{EnV}(\widehat{\mathbf{C}}^1) = \{V_4, V_5\}$  and  $\mathbf{ExV}(\widehat{\mathbf{C}}^1) = \{V_2, V_3\}$ ; similarly,  $V_6$  is directly caused by  $V_3$ ,  $V_4$ , and  $V_5$  with respect to  $\mathbf{EnV}(\widehat{\mathbf{C}}^2) = \{V_6\}$  and  $\mathbf{ExV}(\widehat{\mathbf{C}}^2) = \{V_3, V_4, V_5\}$ . We can also read off *transitive* causal relations, e.g.,  $V_3$  is an indirect cause of  $V_7$  because there is a directed path from  $V_3$  to  $V_7$ . However, the causal relation between  $V_4$  and  $V_5$  is undefined, since they are in a strongly coupled component.

### 3. The theoretical framework for interactive model building

We develop an interactive system, IMAGeNIE, that assists in building graphical decision models in causal form. IMAGeNIE includes three knowledge structures: *mechanism libraries*, which holds domain knowledge expressed as causal mechanisms, *model building workspace*, which serves as a blackboard for model compositions, and *graphical decision models*, which represent decision problems at hand. IMAGeNIE functions as a modeling facilitator that helps structuring a decision problems by means of knowledge provided by *domain experts*. Because a user of IMAGeNIE can be all three: a domain expert, a knowledge engineer, and a decision maker, we will refer to the users of IMAGeNIE as *model builders*. 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 a *hierarchical navigation interface* to locate the mechanisms of interest and to 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 *aggregate mechanisms* as the model building process unfolds.

The mechanism libraries are organized as a hierarchical system that consists of subsystems and causal mechanisms as its

fundamental building elements. The hierarchical approach not only assists model builders in expressing their domain knowledge in cognitively meaningful units but also helps them to access stored mechanisms easily. The approach is similar to type-hierarchy in Koller and Pfeffer (1997) and Laskey and Mahoney (1997) but without imposing the inheritance constraint, since knowledge can be possibly organized hierarchically from different perspectives. The fundamental knowledge units in mechanism libraries are mechanisms, represented as structural equations. Users can specify structural equations in their implicit or explicit functional forms, such as algebraic functions, conditional probability tables, truth tables, value/utility tables, and choice tables. While most mechanisms may be described in one, perhaps their only, mode of operation, some mechanisms will be described in different modes of operation. This is because causal relations among variables are determined by the context, i.e., a system in which they are embedded.

To aid the process of model building, we define the *manipulability* and *observability* properties for each variable in mechanism libraries. A variable is *manipulable* if it can be manipulated directly, i.e., if its value can be set directly, by forces outside of the model. In an influence diagram, for example, all decision variables are manipulable. A variable is *non-manipulable* if its value has to be derived from the model. For example, the mechanism describing the intensity of sunshine ( $S$ ) and our tendency to wear sunglasses ( $SG$ ) can be represented by a structural equation  $f(S, SG) = 0$ . We define  $SG$  as manipulable variables in the modeling domain. The intensity of sunshine  $S$  is determined by the external force of nature. Wearing sunglasses ( $SG$ ) can be controlled by us. 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, we can have  $g(S) = 0$ , which describes the intensity of sunshine, together with  $f(S, SG) = 0$  to model the situation where we wear our sunglasses because of the sunshine ( $S \rightarrow SG$ ). In this example, we have  $S$  as an exogenous variable, but not  $SG$ . However,  $SG$  is still manipulable in our modeling domain, since we can always decide to wear sunglasses regardless of the intensity of sunshine. In such case, we 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 necessarily 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. A variable is *observable* if we can observe or measure its value directly. An unobservable variable is a hidden variable, (e.g., a disease) which has to be inferred through observable variables (e.g., symptoms). 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.

#### 4. Causal ordering in under-constrained systems

Model construction, in IMAGeNIE, is a reflection of molders' problem structuring. 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 an acyclic directed graph, as the causal interactions are not completely determined until a mod-

el is self-contained. However, it is desirable to have a graphical representation of under-constrained models throughout model construction, as the graphical representation can help modelers in identifying focus of modeling and commitments of the outside influences. We extend Simon's causal ordering algorithm to explicate the causal ordering that has been identified in under-constrained models. In addition, we propose a graphical representation of under-constrained models.

In order to formalize extended causal ordering, we prove the following theorem.<sup>2</sup>

**Theorem 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.*

Given Theorem 1, we 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, we know that  $\mathbf{E}$  is self-contained. If  $\mathbf{E}'$  is not empty and no more self-contained structures can be identified, we know that  $\mathbf{E}$  is under-constrained and we call the final  $\mathbf{E}'$  the *derived strictly under-constrained structure*.

**Definition 5** (*strictly under-constrained structure*). An under-constrained structure is strictly under-constrained if it does not contain any self-contained structures.

**Theorem 2.** *A structure  $\mathbf{E}$  is under-constrained if and only if it contains a derived strictly under-constrained structure in  $\mathbf{E}$ .*

Based on Theorem 2, we present the extended causal ordering algorithm (ECO) as follows. The input of the algorithm is a structure matrix  $\mathbf{E}$ . The output is a graph  $G(\mathbf{E}) = \langle \mathbf{N}, \mathbf{A}, \mathbf{A} \rangle$  where  $\mathbf{N}$  represents  $\mathbb{V}(\mathbf{E})$ ,  $\mathbf{A}$  is a set of directed arcs, and  $\mathbf{A}$  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 depict the causal relations and relevant relations encoded in the strictly under-constrained subset, if there remains one.

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 overlapping nodes. Undirected arcs present relevant but undetermined causal relations among variables so that the modeler can focus on clarifying the mechanisms governing these variables to make a model self-contained.

#### 5. Bipartite graph matching and causal ordering

The ECOA presented in Section 4 and COA discussed in Section 2 are 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. Nayak's work focused on automated modeling of physical systems and 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

<sup>2</sup> Due to space limitation, we refer readers to Lu (2003) for details of the proofs of this and other theorems and lemmas reported in this paper.

(1987), who focused on 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 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 this section, we present worst-case polynomial time algorithms for both COA and ECOA, based on bipartite graphs.

We first present the algorithm  $COA_{BGM}$  which is the polynomial time algorithm of COA using bipartite graph matching. The algorithm first represents a structure  $\mathbf{E}$  as a bipartite graph  $BG(\mathbf{E}) = \langle \mathbf{N}_E, \mathbf{N}_V, \mathbb{A} \rangle$ , which 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  $V_i \in \mathbf{V} = \mathbb{V}(\mathbf{E})$ , 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  $V_j \in \mathbb{V}(e_i)$ . 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 vice versa) with respect to a matching  $\mathbb{A}_m \subseteq \mathbb{A}$ , if  $(N_{e_i}, N_{V_j}) \in \mathbb{A}_m$ . A matching is complete if and only if each node in the graph is covered by an arc in the matching (Even, 1979). Such complete matching for  $BG(\mathbf{E})$  resembles one-to-one mapping in the causal ordering for a self-contained structure  $\mathbf{E}$ . In other words, each arc  $(N_{e_i}, N_{V_j})$  in  $\mathbb{A}_m$  can be interpreted as a direct dependency relation where  $\mathbb{V}(e_i) \setminus V_j$  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  $\mathbf{E}$  by  $\mathbb{A}_m$ . However, there might be more than one complete matching for a self-contained structure  $\mathbf{E}$ . Nayak (1994) proved that there exists an unique transitive closure for all possible complete matchings for a self-contained structure  $\mathbf{E}$ . The algorithm, therefore, identifies the strongly connected components in  $DG(\mathbf{E})$ , and interpreting variables within such strongly connected component as interdependent in  $DG(\mathbf{E})$ . The algorithm finally outputs the modified acyclic graph  $DG(\mathbf{E})$  as  $G(\mathbf{E})$  by grouping variables in a strongly coupled component.  $COA_{BGM}$  is worst-case polynomial because we can find the complete matching using network-flow technique with time complexity  $O(\sqrt{na})$ , where  $n = |\mathbf{N}_E| = |\mathbf{N}_V|$  and  $a = |\mathbb{A}|$  (Papadimitriou and Steiglitz, 1982).

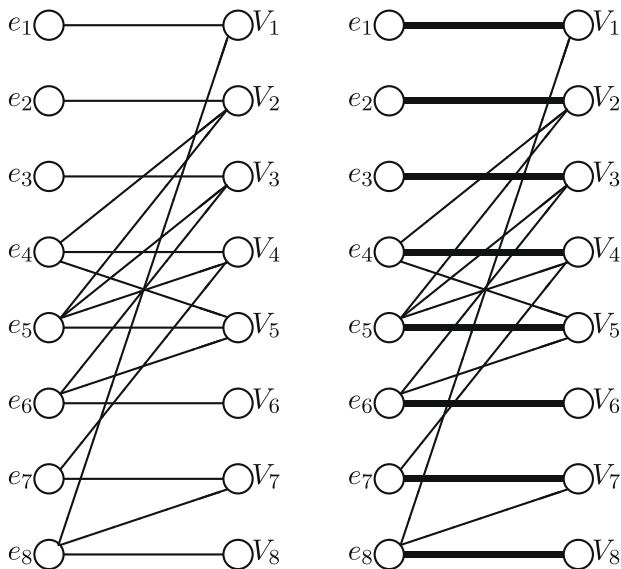


Fig. 2. A bipartite graph (left) and its complete matching (right).

**Example 2.** Consider applying the  $COA_{BGM}$  algorithm on the structure in Example 1.  $COA_{BGM}$  first creates the corresponding bipartite graph of the structure as shown in Fig. 2(left). Next,  $COA_{BGM}$  computes the complete matching as shown in Fig. 2(right), where each bold arc depicts an arc in the matching. Third,  $COA_{BGM}$  creates a directed graph, shown in Fig. 3(left), 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 in Fig. 3(right).

A maximum matching in a bipartite graph is a matching with maximum cardinality. A complete matching is by definition a maximum matching. We prove the following theorem to show that complete matching can be used to determine if a structure  $\mathbf{E}$  is self-contained or under-constrained.

**Theorem 3.** A structure  $\mathbf{E}$  is self-contained, if there is a complete matching in its bipartite graph  $BG(\mathbf{E})$ , and is under-constrained otherwise.

**Proof.** According to Hall's theorem (Even, 1979, pp. 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). Therefore, if there is 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). Therefore, the  $BG(\mathbf{E})$  of an under-constrained structure  $\mathbf{E}$  cannot have a complete matching according to Hall's theorem.  $\square$

Given Theorem 3 and the fact that a complete matching is also a maximum matching, we can apply an algorithm finding maximum matching to the  $BG(\mathbf{E})$  of a given structure  $\mathbf{E}$ . If we find a complete matching, then the structure  $\mathbf{E}$  is self-contained and the identified complete matching can be used to construct the causal graph; otherwise, we have a maximum matching for the under-constrained structure  $\mathbf{E}$ .

Next, we show how to derive the extended causal ordering from the maximum matching of  $BG(\mathbf{E})$ . We first denote the set of matched nodes in any matching  $\mathbb{A}_m$  as  $\mathbf{N}_{E_m} \subseteq \mathbf{N}_E$  and  $\mathbf{N}_{V_m} \subseteq \mathbf{N}_V$ , and the set of unmatched nodes as  $\mathbf{N}_{E_m} \subseteq \mathbf{N}_E$  and  $\mathbf{N}_{V_m} \subseteq \mathbf{N}_V$ . We prove the following theorem to show that given any under-constrained structure  $\mathbf{E}$ , all  $\mathbf{N}_E$  are matched in any maximum matching  $\mathbb{A}_m$  of  $BG(\mathbf{E})$ .

**Theorem 4.** Any maximum matching  $\mathbb{A}_m$  for the  $BG(\mathbf{E})$  of an under-constrained structure  $\mathbf{E}$  has all  $\mathbf{N}_E$  matched ( $\mathbf{N}_{E_m} = \emptyset$ ) and some  $\mathbf{N}_V$  unmatched ( $\mathbf{N}_{V_m} \neq \emptyset$ ).

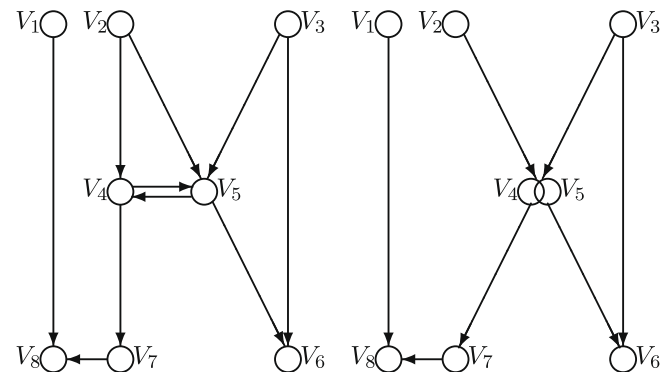


Fig. 3. The directed graph (left) and causal graph (right) created by  $COA_{BGM}$ .

Given Theorem 4, we assign  $\mathbf{Y}$  as  $\mathbf{N}_{V_m}$  and  $\mathbf{X} \subseteq \mathbf{N}_{E_m}$  as those adjacent to  $\mathbf{Y}$  in  $\mathbb{A}$ . We propose the following procedure to identify the strictly under-constrained subset:

- (1)  $\mathbb{A}_{m'} := \mathbb{A}_m$
- (2) **do**
- (3) Identify  $\mathbf{Z} \subseteq \mathbf{N}_{V_m}$  that are matched to  $\mathbf{X}$  in  $\mathbb{A}_{m'}$ .
- (4) Let  $\mathbb{A}_r \subseteq \mathbb{A}_{m'}$  denote the set of arcs that match  $\mathbf{Z}$  and  $\mathbf{X}$ .
- (5)  $\mathbb{A}_{m'} := \mathbb{A}_{m'} \setminus \mathbb{A}_r$ . Consequently,  $\mathbf{Z} \subseteq \mathbf{N}_{V_{m'}}$  and  $\mathbf{X} \subseteq \mathbf{N}_{E_{m'}}$ .
- (6)  $\mathbf{Y} := \mathbf{Z}$ ; Identify  $\mathbf{X} \subseteq \mathbf{N}_{E_{m'}}$  that are adjacent to  $\mathbf{Y}$  in  $\mathbb{A}$ .
- (7) **until**  $\mathbf{X} = \emptyset$ .

Given a finite under-constrained structure  $\mathbf{E}$ , the procedure will stop and return the modified  $\mathbb{A}_{m'}$ . If  $\mathbb{A}_{m'} = \emptyset$ , then  $\mathbf{E}$  is strictly under-constrained. If  $\mathbb{A}_{m'} \neq \emptyset$ , then  $\mathbf{N}_{E_{m'}}$  represents a self-contained structure  $\mathbf{E}_{m'}$  and  $\mathbf{N}_{E_{m'}}$  represents the derived strictly under-constrained structure  $\mathbf{E}_{m'}$ . We prove the following theorem to show that the procedure will result in the same  $\mathbf{N}_{E_{m'}}$  and  $\mathbf{N}_{E_{m'}}$ .

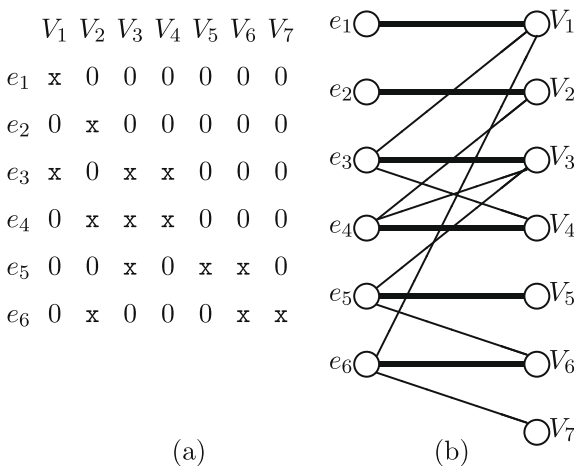
**Theorem 5.** Any maximum matching  $\mathbb{A}_m$  for the BG( $\mathbf{E}$ ) of an under-constrained structure  $\mathbf{E}$  has the same  $\mathbf{N}_{E_{m'}}$  and  $\mathbf{N}_{E_{m'}}$  with respect to the modified matching  $\mathbb{A}_{m'}$ .

Given Theorem 3–5, we have the extended causal ordering algorithm based on maximum matching  $ECO_{BGM}$ , which is a worst-case polynomial time algorithm since it takes polynomial time to find maximum matching and linear time to derive the modified matching. We remark that not all modified matchings  $\mathbb{A}_{m'}$  are the same given different maximum matchings  $\mathbb{A}_m$  of an under-constrained structure  $\mathbf{E}$ . If there is no strongly-coupled component in  $\mathbf{E}_{m'}$  with respect to  $\mathbb{A}_{m'}$ , any maximum matching  $\mathbb{A}_m$  results in the same modified matching  $\mathbb{A}_{m'}$ .

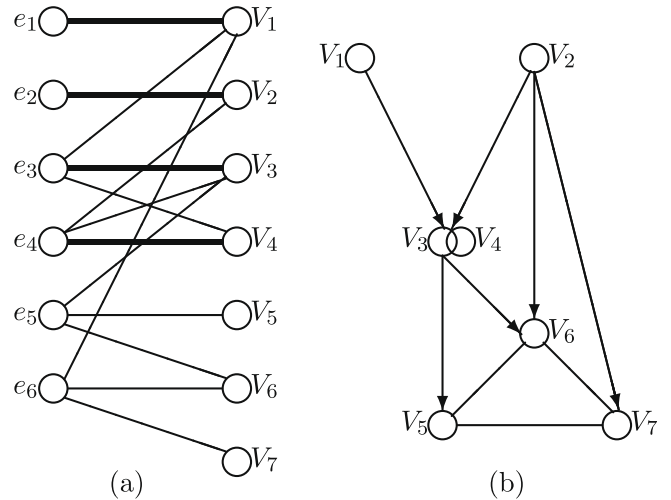
**Example 3.** Consider applying the  $ECO_{BGM}$  algorithm to the structure in Fig. 4a.  $ECO_{BGM}$  first construct the corresponding bipartite graph of the structure as shown in Fig. 4b. Next,  $ECO_{BGM}$  computes the maximum matching as shown in Fig. 4b, where each bold arc depicts an arc in the matching.  $ECO_{BGM}$  then modifies the maximum matching as shown in Fig. 5a.  $ECO_{BGM}$  finally creates the graph, as shown in Fig. 5b, that depicts causal and relevant relations for Fig. 4a.

## 6. Interactive modeling process

The modeling process usually starts with the value variable in the spirit of value-focused thinking (Keeney, 1992). IMAgeNIe sup-



**Fig. 4.** (a) The bipartite graph representation of an under-constrained structure  $\mathbf{E}$ . (b) A maximum matching of the bipartite graph is depicted as bold arcs.



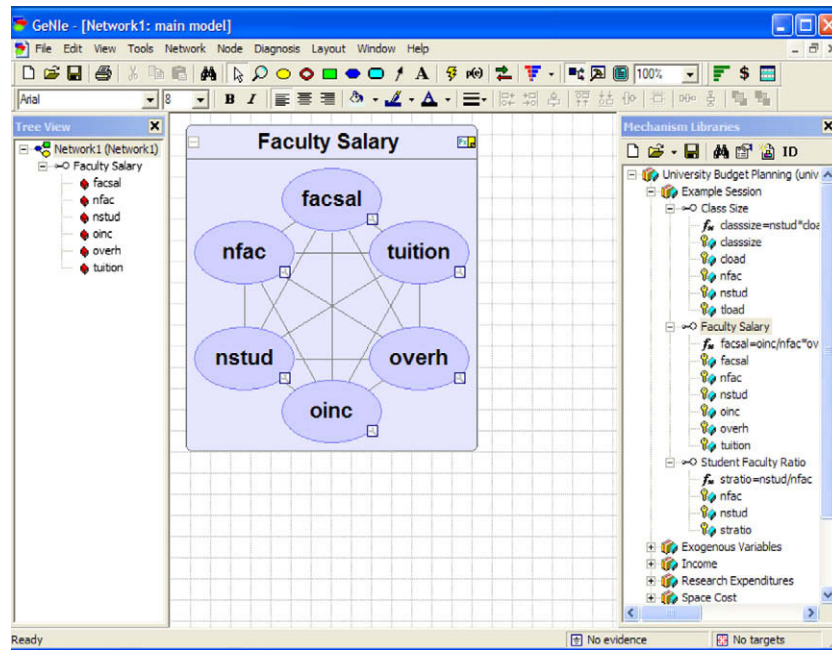
**Fig. 5.** (a) The modified matching of Fig. 4b.  $ECO_{BGM}$  outputs the graph in (b) to depict causal and relevant relations among variables in the under-constrained structure in Fig. 4a.

ports interactive browsing of mechanisms related to their focus variables. Selected mechanisms are brought into the workspace and aggregated with the existing mechanisms. However, we encourage users to focus on one variable and add relevant mechanisms one at a time. This amounts to focusing on a variable of interest and explaining or observing it in terms of its underlying mechanisms. The model builders repeat the process iteratively until the model is requisite. In other words, it is the model builders who decide about the model's level of granularity and when to stop. The system only plays the passive role of a facilitator: assisting in searches for relevant mechanisms, indicating the possible mechanisms to aggregate, selecting the manipulable variables, and showing the status of each variable and (causal or relevant) relations among variables in the workspace.

A model normally starts from an under-constrained model and ends up being 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 model becomes over-constrained, because the number of equations is greater than the number of variables. IMAgeNIe allows a model to be under-constrained or self-contained at any stage of the model development, but not to be over-constrained. When a model becomes over-constrained, IMAgeNIe presents a list of mechanisms that are currently in the model and ask the user to release one of them.

We describe a user's interaction with the system by means of a simple practical example. The University Performance Budget Planning Model (Simon et al., 2000) is composed of 38 nonlinear structural equations that describe interactions among 88 variables. These structural equations are divided into seven subsystems: Teaching Operations, Teaching Expenditures, Research Expenditures, Income, Space Cost, Total Expense, and Surplus. We show below how IMAgeNIe assists a model builders in building a simplified university budget model from structural equations captured in a mechanism libraries.

Suppose Tom, an officer of the budget planning office, would like to investigate consequences of an increase in the faculty salary. He starts up IMAgeNIe and loads the university mechanism library. 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*), num-

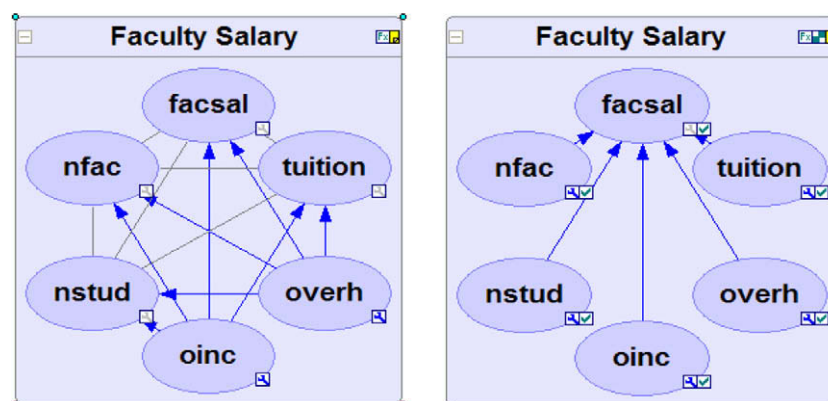


**Fig. 6.** Tom selects a mechanism from libraries (right pane) into workspace. IMAGeNIe generates a graphical model (middle pane) and variables in the model are also displayed in the network tree view (left pane).

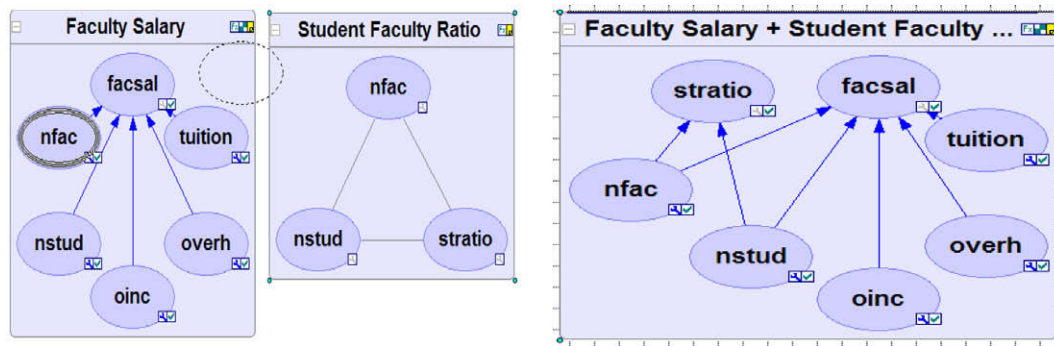
ber of faculty (*nfac*), and overhead (*overh*), as  $facsal = (oinc + tuition * nstud) / (nfac * (1 + overh))$ . He selects the mechanism by clicking and dragging it into the model building workspace. The extended causal ordering module generates a corresponding graph as shown in Fig. 6 and the variables 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 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 Fig. 7). Tom then identifies the student–faculty ratio mechanism ( $stratio = nstud/nfac$ ) 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. Tom then integrates these two mechanisms by selecting the node *number of students* (*nstud*) from one (source) mechanism box and dragging it over to the other (target) mechanism box. When the cursor enters the target mecha-

nism box, IMAGeNIe searches over the variable identifiers in the target mechanism box and highlights the variable with identical identifiers. Tom then releases the mouse cursor and IMAGeNIe automatically integrates other variables with the same identifiers in both mechanism boxes and integrates two mechanism boxes into one box with two mechanisms (see Fig. 8). Tom then brings in the mechanism describing the interactions among variables: *class size* (*classsize*), *number of students* (*nstud*), *class load* (*cload*), *number of faculty* (*nfac*), and *teaching load* (*tload*), as  $classsize = (nstud * cload) / (nfac * tload)$ . He aggregates 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 (Fig. 9).

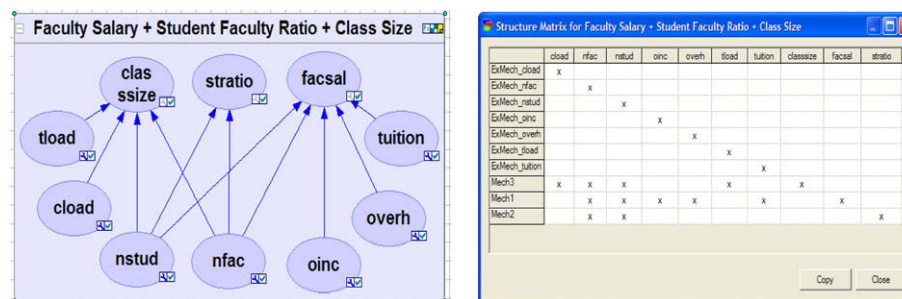
Note that IMAGeNIe has effectively assisted Tom in focusing from one value variable (*facsal*) to other relevant variables. In particular, IMAGeNIe assisted Tom in considering other decision criteria by examining what other variables may be effected by the causes variables. For example, by exploring mechanisms relevant to *nstud* and *nfac*, Tom identified two mechanisms related to decision criteria



**Fig. 7.** Tom designates the variables *number of students* (*nstud*), *number of faculties* (*nfac*), *tuition* (*tuition*), *overhead* (*overh*), and *other income* (*oinc*) as exogenous (left). IMAGeNIe generates the corresponding graph (right).



**Fig. 8.** When Tom drags *nfac* from the *student–faculty ratio* mechanism box entering into the *faculty salary* mechanism box, the variable with the same identifier is highlighted (the triple lines around *nfac* at left). IMAGeNIE performs the aggregate operations for *nstud* and *nsfac*, as soon as model builders drop the *nsfac* over the highlighted *nsfac*, and generate the corresponding graph of aggregated mechanisms (right).



**Fig. 9.** A simplified university model that contains three core mechanisms and seven value assignment equations. The causal graph of the model is shown on the left; and the structure matrix of the model is shown on the right.

*faculty–teacher ratio* and *class size*. In other words, IMAGeNIE has effectively helped Tom to build a multi-criteria graphical decision model. Furthermore, IMAGeNIE can supports trade-off analysis by plotting the decision surface of an effect variable, given the value ranges of cause variables (see Fig. 10).

Finally, Tom makes the *facsal* exogenous to see the consequences of an increase in the faculty salary. Since the system is self-contained, IMAGeNIE searches for relevant mechanisms that could be released to ensure the system remain self-contained.<sup>3</sup> The dialogue box in Fig. 11 shows that Tom chooses to release *tuition* as an exogenous variable. IMAGeNIE then restructures the graph (Fig. 11) to reflect such changes. In other words, increasing the faculty salary will result in the increase of tuition, while leaving other mechanisms intact.

## 7. Evaluation

We compared the effectiveness of IMAGeNIE in constructing causal models to the effectiveness of a plain graphical modeling environment and to a spreadsheet in a within-subjects design. The description below is brief because of space limitations. Lu (2003) contains a detailed description of the experiments:

- (1) **Subjects:** Subjects were 40 graduate students who either were taking the course *Decision Analysis and Decision Support Systems* offered by the second author in the Spring 2003 semester at the School of Information Sciences or who had taken this course in the past. All subjects had received basic training in decision analysis in the course and were familiar

with the graphical modeling tools such as GENIE. All subjects were volunteers and were compensated for their participation by a small course credit and candies.

- (2) **Materials:** Each subject received the following: (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, and (3) a questionnaire to fill out after the experiment. We asked the subjects to rate the usefulness of the three systems for solving each task case on an 11-point scale ranging from *useless* (0) to *extremely useful* (10). We then asked for subjects' open-ended comments on using different systems in solving each task case. We chose four mechanisms involving nine variables from the domain of stock market investment and constructed two test cases. Each of the subjects was fairly familiar with this domain. In Task I, the subjects were asked to solve a problem by constructing models using IMAGeNIE, EQGeNIE and EXCEL. In Task II, the subjects were asked to solve another problem by changing the structure of the models constructed in Task I. In order to make the comparison fair to EQGeNIE and EXCEL, we made sure that the gold standard solutions of the two cases were directed acyclic graphs, i.e., they contained no strongly-coupled components. To ensure that subjects were familiar with the functionalities needed for solving cases in IMAGeNIE, EQGeNIE and EXCEL, we first gave a brief demo of each of the systems. Then we assisted each subject in constructing models for an example case and made sure that each subject had sufficient confidence and skill in working with the systems.
- (3) **Evaluation criteria:** *Effectiveness:* We measured structural discrepancy between the constructed models and the gold standard models. While we have formalized the distance measure, space limitations do not allow us to present it in

<sup>3</sup> Due to space limitation, we refer readers to Lu (2003) for theoretical aspects of support for changes in structure.

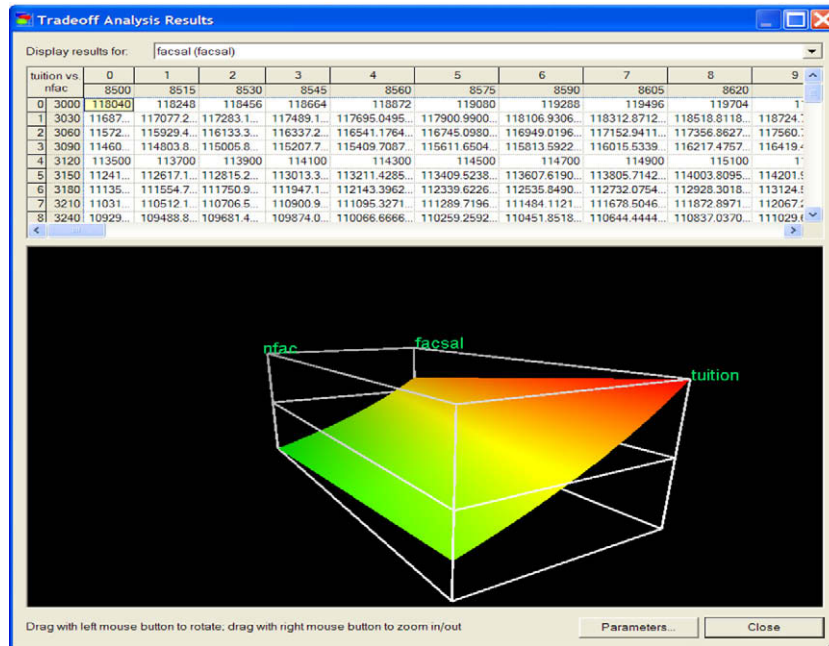


Fig. 10. IMAGeNIE supports trade-off analysis by showing the decision surface of effect variable *facsal* given the cause variables *nstud* and *nfac*.

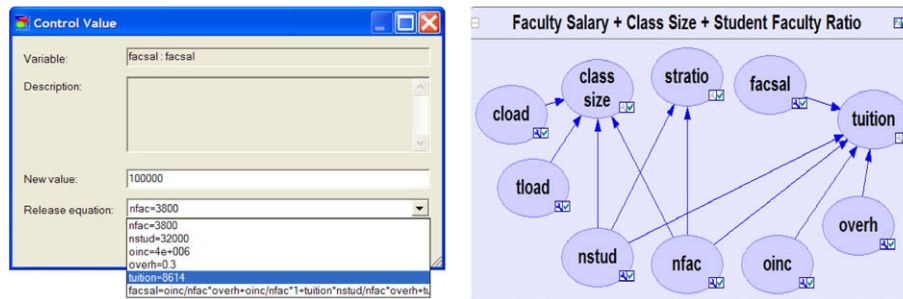


Fig. 11. Tom makes *facsal* exogenous by controlling its value. IMAGeNIE requires that one of mechanisms be released to ensure the system is self-contained. Tom releases the *tuition* mechanism (left). IMAGeNIE restructures the system and generates the corresponding graph (right).

detail. Simply speaking, structural discrepancy consisted of the count of arcs omitted or drawn incorrectly between the solution and the gold standard. Although models constructed by EXCEL system were not in a graphical form, we converted them into directed graphs by reading off the dependency relations specified in cells. *Efficiency*: For each subject  $i$  and task  $j$ , we recorded the time taken to construct models. *Usefulness*: We collected the subjective evaluation of usefulness of each of the systems as part of the questionnaire given at the conclusion of the experiment.

- (4) **The Results:** Of the 40 subjects who participated in the experiment, four dropped out for personal reasons, leaving 36 usable results. The effectiveness measure did not show any significant difference between the three systems ( $p = 0.832$  for Task I and  $p = 0.289$  for Task II, ANOVA). In terms of efficiency, it looked like EXCEL was a clear winner in Task I ( $p = 1.039 \times 10^{-7}$  ANOVA) but IMAGeNIE was a clear winner in Task II ( $p = 1.191 \times 10^{-9}$  ANOVA). Similarly, usefulness measure showed significant results (the measures were different for the three systems at  $p = 0.00099$  for Task I and  $p = 1.621 \times 10^{-9}$  for Task II, ANOVA).
- (5) **Discussion:** The subjects found IMAGeNIE an efficient and useful system for causal model construction. We did not find any significant differences among the effectiveness of model

construction using different systems. While EXCEL outperformed IMAGeNIE and EQGeNIE in terms of the average time taken to solve the problems, it might be explained by our subjects' a priori familiarity with it. The fact that IMAGeNIE outperformed EXCEL and EQGeNIE on Task II shows that IMAGeNIE is helpful in the task of changes in structure.

## 8. Conclusion

Supporting model structuring is one of the best ways to improve the ultimate quality of advice suggested by decision models. Our framework supports building decision models by viewing the process of model structuring as assembling of individual mechanisms into a self-contained system. In such process, we often encounter under-constrained systems, since not all relevant mechanisms may have been brought into the model. We extend Simon's causal ordering algorithm to explicate causal relations in under-constrained systems. Graphs encoding under-constrained systems, in turns, help model builders exploring relevant mechanisms and specifying exogenous variables. Built upon our framework and the extended causal ordering, IMAGeNIE functions as a model facilitator that assists model builders (domain experts, knowledge engineers, and decision makers) in selecting, aggregating, exploring, controlling, and extracting mechanisms for decision problems

at hand. Furthermore, the resulting models are guaranteed to be causal, as defined by mechanism-based view of causality, if the underlying structural equations reflect causal mechanisms of the modeled problem. In this paper, we have also briefly shown how IMAGeNIE can support for multi-criterion decision problems, trade-off analysis, and changes in structure. In Lu (2003), we further showed how IMAGeNIE can assist the problem of *search for opportunities* by an automatic designation of those manipulable variable to be decision nodes and, in effect, search for those variables that can give the largest effect on the utility function over outcomes of a priori specified value nodes.

## Acknowledgements

This work was supported by the Air Force Office of Scientific Research grant FA9550-06-1-0243 and by Intel Research. We thank anonymous reviewers for their suggestions that led to improvements of the paper. IMAGeNIE is based on GeNIE, a development environment for reasoning in graphical probabilistic models developed at the Decision Systems Laboratory, University of Pittsburgh, and available at <http://genie.sis.pitt.edu/>.

## References

- Breese, J.S., Goldman, R.P., Wellman, M.P., 1994. Introduction to the special section on knowledge-based construction of probabilistic and decision models. *IEEE Transactions on Systems, Man and Cybernetics*, Part A 24 (11), 1577–1579.
- Cooper, G.F., Herskovits, E., 1991. A Bayesian method for constructing Bayesian belief networks from databases. In: *Proceedings of the UAI-91*. Morgan Kaufmann Publishers, San Mateo, CA, pp. 86–94.
- Diez, F.J., 1993. Parameter adjustment in Bayes networks. *The generalized noisy OR-gate*. In: *Proceedings of the UAI-93*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 99–105.
- Druzdzel, M.J., 1992. Probabilistic Reasoning in Decision Support Systems: From Computation to Common Sense. Ph.D. Thesis, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA.
- Druzdzel, M.J., Oniško, A., 2008a. The impact of overconfidence bias on practical accuracy of Bayesian network models: an empirical study. In: *Working Notes of the 2008 Bayesian Modelling Applications Workshop*, Special Theme: How Biased Are Our Numbers, Part of the Annual Conference on Uncertainty in Artificial Intelligence (UAI-2008), Helsinki, Finland, 9 July 2008.
- Druzdzel, M.J., Oniško, A., 2008b. Are Bayesian networks sensitive to precision of their parameters. In: Wierzbchoń, S.T., Opotek, M.K., Michalewicz, M. (Eds.), *Intelligent Information Systems XVI*, *Proceedings of the International IIS'08 Conference*, Academic Publishing House EXIT, Warsaw, Poland, June 2008, pp. 35–44.
- Druzdzel, M.J., Simon, H.A., 1993. Causality in Bayesian belief networks. In: *Proceedings of the UAI-93*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 3–11.
- Druzdzel, M.J., van der Gaag, L.C., 2000. Building probabilistic networks: “Where do the numbers come from?” guest editors’ introduction. *IEEE Transactions on Knowledge and Data Engineering* 12 (4), 481–486.
- Even, S., 1979. *Graph Algorithms*. Computer Science Press, Potomac, MD.
- Goldberger, A.S., 1972. Structural equation methods in the social sciences. *Econometrica* 40 (6), 979–1001.
- Haavelmo, T., 1944. The probability approach in econometrics. *Econometrica* 12 (Issue Supplement), 1–115.
- Heckerman, D., 1990. Probabilistic similarity networks. *Networks* 20 (5), 607–636.
- Henrion, M., 1989. Some practical issues in constructing belief networks. In: Kanal, L., Levitt, T., Lemmer, J. (Eds.), *Uncertainty in Artificial Intelligence*, vol. 3. Elsevier Science Publishing Company Inc., New York, NY, pp. 161–173.
- Holtzman, S., 1989. *Intelligent Decision Systems*. Addison-Wesley, Reading, MA.
- Howard, R.A., Matheson, J.E., 1981. Influence diagrams. In: Howard, R.A., Matheson, J.E. (Eds.), *The Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA, pp. 719–762.
- Iwasaki, Y., Simon, H.A., 1994. Causality and model abstraction. *Artificial Intelligence* 67 (1), 143–194.
- Keeney, R.L., 1992. *Value-focused Thinking: A Path to Creative Decision Making*. Harvard University Press, Cambridge, MA.
- Koller, D., Pfeffer, A., 1997. Object-oriented Bayesian networks. In: *Proceedings of the UAI-97*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 302–313.
- Laskey, K.B., 2004. MEBN: A Logic for Open-world Probabilistic Reasoning. Technical report, George Mason University, Department of Systems Engineering and Operations Research.
- Laskey, K.B., Mahoney, S.M., 1997. Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In: *Proceedings of the UAI-97*, Providence, RI, pp. 334–341.
- Lu, T.-C., 2003. Construction and Utilization of Mechanism-based Causal Models. Ph.D. Thesis, University of Pittsburgh.
- Maaskant, P.P., Druzdzel, M.J., 2008. An ICI Model for opposing influences. In: *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM-08)*. In: Jaeger, M., Nielsen, T.D. (Eds.), Hirtshals, Denmark, September 17–19, 2008, pp. 185–192.
- Mahoney, S.M., Laskey, K.B., 1996. Network engineering for complex belief networks. In: *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 389–396.
- Mingers, J., Rosenhead, J., 2004. Problem structuring methods in action. *European Journal of Operational Research* 152, 530–554.
- Montibeller, G., Belton, V., 2006. Causal maps and the evaluation of decision options – A review. *Journal of the Operational Research Society* 57, 779–791.
- Nadkarni, S., Shenoy, P., 2004. A causal mapping approach to constructing Bayesian networks. *Decision Support Systems* 38, 259–281.
- Nayak, P.P., 1994. Causal approximations. *Artificial Intelligence* 70 (1–2), 1–58.
- Oniško, A., Druzdzel, M.J., 2003. Effect of imprecision in probabilities on the quality of results in Bayesian networks: an empirical study. In: *Working Notes of the European Conference on Artificial Intelligence in Medicine (AIME-03) Workshop on Qualitative and Model-based Reasoning in Biomedicine*, Protaras, Cyprus, 19 October, 2003, pp. 45–49.
- Papadimitriou, C.H., Steiglitz, K., 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall Inc.
- Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Mateo, CA.
- Pearl, J., Verma, T.S., 1991. A theory of inferred causation. In: Allen, J., Fikes, R., Sandewall, E. (Eds.), *KR-91, Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*. Morgan Kaufmann Publishers Inc., San Mateo, CA/Cambridge, MA, pp. 441–452.
- Pfeffer, A., Koller, D., Milch, B., Takusagawa, K.T., 1999. SPOOK: A system for probabilistic object-oriented knowledge representation. In: *Proceedings of the UAI-99*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 541–550.
- Phillips, L.D., 1984. A theory of requisite decision models. *Acta Psychologica* 56, 29–48.
- Poole, D., 2003. First-order probabilistic inference. In: *International Joint Conference on Artificial Intelligence*. Acapulco, Mexico, pp. 985–991.
- Pradhan, M., Henrion, M., Provan, G., Del Favero, B., Huang, K., 1996. The sensitivity of belief networks to imprecise probabilities: An experimental investigation. *Artificial Intelligence* 85 (1–2), 363–397.
- Serrano, D., Gossard, D., 1987. Constraint management in conceptual design. In: *Sriram, D., Adey, R. (Eds.), Knowledge Based Expert Systems in Engineering: Planning and Design*. Computational Mechanics Publications, Southampton, pp. 211–224.
- Simon, H.A., 1953. Causal ordering and identifiability. In: Hood, W.C., Koopmans, T.C. (Eds.), *Studies in Econometric Method*. Cowles Commission for Research in Economics. Monograph No. 14. John Wiley and Sons Inc., New York, NY, pp. 49–74 (Ch. III).
- Simon, H.A., 1979. The meaning of causal ordering. In: Merton, R.K., Coleman, J.S., Rossi, P.H. (Eds.), *Qualitative and Quantitative Social Research: Papers in Honor of Paul F. Lazarsfeld*. The Free Press, A Division of Macmillan Publishing Co. Inc., pp. 65–81. Ch. 8.
- Simon, H.A., Kalagnanam, J.R., Druzdzel, M.J., 2000. Performance Budget Planning: The Case of a Research University. Unpublished manuscript.
- Simon, H.A., Rescher, N., 1966. Cause and counterfactual. *Philosophy of Science* 33 (4), 323–340.
- Spirtes, P., Glymour, C., Scheines, R., 1993. *Causation, Prediction, and Search*. Springer-Verlag, New York.
- Srinivas, S., 1993. A generalization of the noisy-OR model. In: *Proceedings of the UAI-93*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 208–215.
- von Winterfeldt, D., Edwards, W., 1986. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge.
- Wold, H., 1954. Causality and econometrics. *Econometrica* 22 (2), 162–177.