# CONSTRUCTIVE INDUCTION IN STRUCTURAL DESIGN

by

*T. Arciszewski*
*E. Bloedorn*
*R. S. Michalski*
*M. Mustafa*
*J. Wnek*

In Reports of Machine Learning and Inference Laboratory, Center for Artificial Intelligence, George Mason University, December 1992.

# CONSTRUCTIVE INDUCTION IN ENGINEERING DESIGN

T. Arciszewski, E. Bloedorn
R.S. Michalski, M. Mustafa and J. Wnek

MLI 92-7

# CONSTRUCTIVE INDUCTION IN STRUCTURAL DESIGN


Tomasz Arciszewski[1], Eric Bloedorn[2], Ryszard S. Michalski[2]
Mohamad Mustafa[1] and Janusz Wnek[2]

**[1] Department of Civil and Environmental Engineering**
**Wayne State University**
**Detroit, MI 48202**

**[2] Artificial Intelligence Center**
**George Mason University**
**Fairfax, VA 22030**



**MLI 92-7**

**December 1992**

# CONSTRUCTIVE INDUCTION IN ENGINEERING DESIGN


## Abstract

This paper discusses the applicability of advanced machine learning methods to problems of conceptual design, and presents a case study on an automated acquisition of design rules for wind bracings in tall buildings. Design rules are determined by generalizing expert-given examples of designs. The generalization methods used in the study are based on the most recent research in machine learning, which concerns the *constructive induction* programs capable of generating new attributes, beyond those provided initially, and automatically formulating decision rules. The process of determining rules consists of three stages: the problem definition, preparation of the examples of designs, and rule induction from examples. The decision rules generated by machine learning programs specify design recommendations, designs to be avoided, typical (standard) designs, and infeasible designs. The learned rules were analyzed and verified. They were evaluated as highly promising and capturing the essential understanding of the wind bracings conceptual design which is typically used in the design process. The results obtained indicate a great promise of the methods of constructive induction in structural design knowledge acquisition.



**Key Words:** Machine Learning, Structural Design Knowledge Acquisition, Constructive Induction, Data-Driven and Hypothesis-Driven Constructive Induction



## Acknowledgements

The authors would like to thank Dr. Yoram Reich of Carnegie Mellon University for his useful comments and suggestions regarding this paper. We also acknowledge the cooperation of Waterloo Engineering Software, Ontario, Canada, which provided SODA used in the preparation of examples.

This research was conducted in the Center for Artificial Intelligence at George Mason University. The Center's research is supported in part by the Defense Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, in part by the Office of Naval Research under Grant No. N00014-91-J-1351, and in part by the National Science Foundation under Grant No. IRI-9020266.

## INTRODUCTION

Machine learning is a scientific discipline concerned with understanding computational principles of learning and discovery, and developing computer systems exhibiting such capabilities. It represents an interdisciplinary research effort that draws upon results from such disciplines as computer science, artificial intelligence, cognitive science, information systems, and cognitive psychology.

Although efforts to build learning machines go back to the early years of computer era, a major progress in understanding how to build practical computer systems with learning capabilities has occurred relatively recently. In the last several years machine learning programs have been experimentally tried in a variety of domains, such as engineering, medicine, agriculture, computer vision, financial decision making, and others. The field has reached a stage of maturity that applying it to engineering is now not only feasible, but may likely bring useful results.

The engineering disciplines are undergoing now a paradigm shift. The analytical paradigm, based on the use of analytical tools, is rapidly becoming insufficient. At the same time, a new paradigm is emerging, founded on the use of knowledge-based decision support tools. However, the development, and subsequent use of such tools requires the acquisition of knowledge in the form of decision rules.

Traditional methods of manual knowledge acquisition are insufficient to deal with complex engineering problems. The progress in using decision support tools has been delayed due to the difficulties of knowledge acquisition. The solution to this problem is *automated knowledge acquisition*, based on the use of learning systems. But this is impossible in engineering today, because of the existing methodological gap between machine learning research in computer science, concerned primarily with the internal workings of learning systems, and the needs of potential users. In this context, there is a significant need to develop a methodology for applying learning systems to problems of automated knowledge acquisition in engineering, which would reflect specific requirements and intricacies of this domain of application.

The present advanced stage of research in the theory of machine learning, together with available computer implementations, justify an investigation into the application of machine learning to structural design. This paper reports the results of an initial feasibility study using advanced methods for machine learning. The methods apply constructive induction, i.e., induction which is not restricted to an initial problem representation. The study was conducted in the Machine Learning and Inference Laboratory at the Center for Artificial Intelligence at George Mason University in Fairfax, Virginia. The examples of optimal structural designs were prepared in the Intelligent Computer Laboratory of the Civil Engineering Department at Wayne State University, Detroit, Michigan.

Some work has already been done on the application of machine learning to conceptual design knowledge acquisition, as briefly reviewed here. Gero (et al. 1989) (Mackenzie and Gero 1987) (McLaughlin and Gero 1987) used an experimental learning system based on the ID-3 algorithm to acquire conceptual design knowledge about architectural design. Reich (1991) developed COBWEB, an experimental incremental learning program for the unsupervised concept learning and used it to acquire design knowledge about bridges (Reich and Fenves 1992). BRIDGER is an extension of COBWEB, developed by Reich (1991) for the same purpose. Reich and Fenves also studied knowledge acquisition about floor-system design in buildings (Reich and Fenves 1988) using Soar, a computer program with learning capabilities. A genetic algorithm was used by Maher (1992 a,b) to develop a learning system for the general purposes of knowledge acquisition about conceptual design. The same algorithm was also used by Grierson and Pak (1992) in a system for the optimization of configurations in the conceptual design of skeletal building structures and by Hajela for structural synthesis (1989). ROUGH, a learning system based on the theory of rough sets (Pawlak 1982, Ziarko 1989), was used by Arciszewski et al. (1987) (Mustafa and Arciszewski 1992) for design knowledge acquisition in the area of structural design of wind bracings in tall buildings. This system uses some form of constructive induction, and the promising results obtained motivated us to study constructive induction in the context of design knowledge acquisition. Garrett and Ivezic (in print) are working on the development of an experimental learning system, NETSYN, based on a connectionist learning approach, for the acquisition of conceptual design knowledge. Also, Adeli and Yeah (1989) used a neural network to learn about engineering design. Milzner and Harbecke (1992) are involved in the development of an experimental learning system, LEAR, based on constructive induction, for learning design knowledge. This system, however, has not been used yet for experiments with actual engineering design examples. Whitehall, Stepp and Lu (1990) (Lu and Chen 1987) are working on machine learning in design knowledge acquisition using several experimental learning systems. The AQ15 and AQ17 learning systems, employing the STAR methodology (Michalski and Stepp 1986), were applied by Arciszewski and Dybala (1992) for learning design rules in the area of conceptual design of wind bracings in steel skeleton structures of tall buildings. Some of the projects reported above were terminated, but the majority are still in progress and should produce results soon, and the feasibility of applying machine learning to design knowledge acquisition will be better understood.

The reported research has two major related objectives: 1) to determine the feasibility of constructive induction applied to knowledge acquisition in structural design, and 2) to determine the performance accuracy of constructive induction-based learning systems in structural design knowledge acquisition. The feasibility of constructive induction in structural design knowledge acquisition was determined, taking into consideration domain relevance and the significance of decision rules, while the performance accuracy of constructive induction-based learning systems was formally determined using two empirical error rates, as proposed in Arciszewski, Dybala and Wnek (1992).

This paper reports the initial results of the feasibility study of constructive induction in structural design. It provides a brief description of the two forms of constructive induction considered, including data-driven and hypothesis-driven constructive induction. The preparation of examples is discussed, and a detailed description of the knowledge acquisition process and its results is given. Conclusions and recommendations for further research are also included.

## CONSTRUCTIVE INDUCTION

### Basic Concepts

*Machine learning* studies mechanisms for creating or improving knowledge or skills by exploiting experience. Since learning is a fundamental characteristic of intelligent behavior, machine learning has been a focus of research in artificial intelligence since the beginnings of AI in the 1950's. The last several years have marked a period of great expansion and diversification of methods and approaches to machine learning. Most of the research has been oriented toward monostrategy methods that apply one primary type of inference and/or computational mechanism. Such methods include, for example, learning decision rules from examples. With the growing understanding of the capabilities and limitations of monostrategy methods, there has been an increasing interest in *multistrategy learning* systems that integrate two or more inference types and/or computational mechanisms. Such systems take advantage of the strengths of different learning strategies, and thus potentially can be applied to a much wider range of practical problems than monostrategy systems (Michalski and Tecuci, to appear 1993). One of the most important concerns of machine learning research is the dependence of learning programs on the initial attributes or predicates used to describe the data. The effectiveness of learning algorithms is very sensitive to the choice of representation space, in particular to the quality of the attributes and predicates used.

*Constructive induction* (CI) is a type of induction in which the formation of a new representation occurs during inductive learning (Michalski 1978). Therefore, instead of generalizing input information in the same original description space, CI methods create a new space in which it may be much easier to construct the appropriate hypothesis. The process of changing the representation space may involve both deductive and inductive inferences, such as abstraction and concretion.

*Abstraction* is a deductive, truth preserving process that reduces the amount of detail in a description of an entity. To do so, it changes the representation to one that uses more abstract concepts, and is more suitable for expressing the properties of the entity relevant to the reasoner's goal. This can be done by elimination of some attributes. *Concretion* is an inductive process that generates additional details about a given entity (Michalski to appear 1993) through the introduction of new or constructed attributes.

From the representational point of view, engineering applications provide very diverse sets of learning problems. They range from applications where concepts are well defined using just a few of the original variables, to applications where concept descriptions require complex transformations and new attributes. The former were already well studied, and are available in the form of domain theories (e.g. Theory of Elasticity, Theory of Plasticity, etc.). In the latter case, regularities are not obvious, due primarily to inadequate domain representations. Inadequacy in the domain representation may occur in three forms: 1) irrelevant attributes, 2) insufficient descriptors (hidden relationships between descriptors), or 3) a combination of the previous two. To deal with these problems, we investigated two methods of constructive induction: 1) data-driven constructive induction (DCI), which constructs new attributes based on input data analysis and application of various mathematical and logical operators, and 2) hypothesis-driven constructive induction (HCI), which builds new attributes by analyzing initially created inductive hypotheses, and detecting patterns in their descriptions.

## Data-Driven Constructive Induction: DCI

Most inductive learning programs perform "selective" induction, that is, they generate descriptions (rules, decision trees, etc.) that involve only attributes selected among those provided in the examples. Thus, if the attributes used in the examples are weakly relevant, the learned descriptions may also be weak. It is possible, however, that although the original attributes may be of poor quality, there exist certain combinations or functions of these attributes that are highly relevant to the problem. For example, suppose there exist two sets of designs of wind bracings in a building described by the number of vertical and horizontal trusses. Sample data are shown in Table 1.

Table 1. Wind Bracings Described by Numbers of Vertical and Horizontal Trusses.

| Class 1 | | Class 2 | |
|---|---|---|---|
| No vertical trusses | No horizontal trusses | No vertical trusses | No horizontal trusses |
| 0 | 2 | 0 | 3 |
| 1 | 1 | 1 | 2 |
| 2 | 0 | 2 | 1 |
| | | 1 | 2 |

The rule which describes the characteristic of each class of building found by a selective program such as AQ14 is fairly complex:

Class1 <:: [Number of Vertical Trusses = 2] and [Number of Horizontal Trusses = 0] or
[Number of Vertical Trusses = 1] and [Number of Horizontal Trusses = 1] or
[Number of Vertical Trusses = 0] and [Number of Horizontal Trusses = 2]

Class2 <:: [Number of Vertical Trusses = 3] or
[Number of Vertical Trusses = 2] and [Number of Horizontal Trusses = 1] or
[Number of Vertical Trusses = 1] and [Number of Horizontal Trusses = 2] or
[Number of Horizontal Trusses = 3]

The complexity of the rule is due to the weakness of the representation. By generating new attributes, the representation is enriched, and the rules constructed can be simpler and also more accurate. Various combinations of attributes using a variety of operations can be calculated. Useful combinations, which in this case involve addition, are kept. In this example, the total number of trusses was calculated and found to be useful. This value was then retained and named Total Number of Trusses. The rules produced using this constructed attribute were:

Class1 <:: [Total Number of Trusses = 2]
Class2 <:: [Total Number of Trusses = 3]

where *Total Number of Trusses* is Number of Vertical Trusses + Number of Horizontal Trusses.

The DCI method is based on the generate and test paradigm. First, all numeric attributes are identified and paired. Then, the operations to be performed on the individual pairs are selected from the list supplied by the user. With the attributes and operation selected, the values for the new attribute are calculated. The discriminatory power of these attribute values is then tested using the Attribute Quality Function (AQF). The AQF is the ratio defined below:

$$AQF = \frac{Unique\ examples}{Total\ number\ of\ examples\ in\ class}$$

where, unique examples are those examples that are not covered by any rule in other classes.

The AQF is calculated for each class and for each attribute. A perfect discriminatory attribute, which alone discriminates one class from all other classes, will have an AQF value of 1. Possible AQF values range from 0 to 1. If a newly constructed attribute exceeds a predefined threshold for quality (quality thresholds are defined for each operator) then this new attribute is added to the list of available attributes and its calculated values are added to the training data.

A number of different operations are available to construct new attributes. These operations can be classified as either binary operators or multi-argument operators (functions). The binary group currently includes the Relational Operator, and a number of mathematical operators, including: Addition, Subtraction (absolute difference), Multiplication, and Integer-Division. Examples of each of these operations on fictitious data are shown in Table 2.

The multi-argument class includes the following functions: Maximum, Minimum, Average, Least-Common, Most-Common, and #VarEQ(x). Except for the latter, these function are self-explanatory. #VarEQ(x) is a function which calculates the number of times the value x appears in an example. For a vector of binary-valued attributes, #VarEQ(1) counts the number of attributes that have a value of 1 in an example of a given class. Examples of these operations are shown in Table 3.

Table 2. Data-Driven Constructive Induction: Binary Operators.

| Operator | Attribute 1 | Attribute 2 | Result |
|---|---|---|---|
| Relation | x | y | 1 if x=y; 2 if x<y; 3 if x>y |
| Addition | 6 | 8 | 14 |
| Subtraction | 6 | 8 | 2 |
| Mulitplication | 6 | 8 | 48 |
| Integer-Division | 6 | 8 | 0 |

Table 3. Data-Driven Constructive Induction: Functional Operators.

| Operator | Attribute 1 | Attribute 2 | Attribute 3 | Result |
|---|---|---|---|---|
| Maximum | 4 | 7 | 4 | 7 |
| Minimum | 4 | 7 | 4 | 4 |
| Average | 4 | 7 | 4 | 5 |
| Most-Common | 4 | 7 | 4 | 4 |
| Least-Common | 4 | 7 | 4 | 7 |
| #VarEQ(4) | 4 | 7 | 4 | 2 |
| #VarEQ(7) | 4 | 7 | 4 | 1 |

The program has a default list of global functions, but allows the user to modify the list to fit the problem at hand. The default list of functions includes maximum, minimum, average, most frequent, least frequent, and #VarEQ(x).

### Hypothesis-Driven Constructive Induction: HCI

The hypothesis-driven constructive induction method extends the capabilities of a selective induction learning algorithm by constructing and using new attributes based on an analysis of the hypotheses generated. It relies on the capability of the selective algorithm to generalize from examples to more general classification rules, given fixed representation space, i.e., a set of attributes and a set of concepts. The HCI method changes the representation space with respect to the set of attributes, or in general, descriptors. The set of concepts, as initially defined by a domain expert, remains unchanged. Figure 1 presents a diagram illustrating the HCI method (Wnek and Michalski 1993).

In the implemented system, the input consists of training examples of one or more concepts, and background knowledge about the attributes used in the examples (a specification of their types and legal value sets). For the sake of simplicity, let us assume that the input consists of positive examples and negative examples of only one concept. If there are several concepts to learn, examples of each concept are taken as positive examples of that concept, and the set theoretical union of examples of other concepts are taken as negative examples of that concept.

The "Split of Examples" module randomly divides positive and negative training examples into primary, P, and secondary, S. The primary training set, P, is used for initial rule learning, the secondary set, S, for an evaluation of intermediate rules, and complete training set, $P \cup S$, is used for the final rule learning.

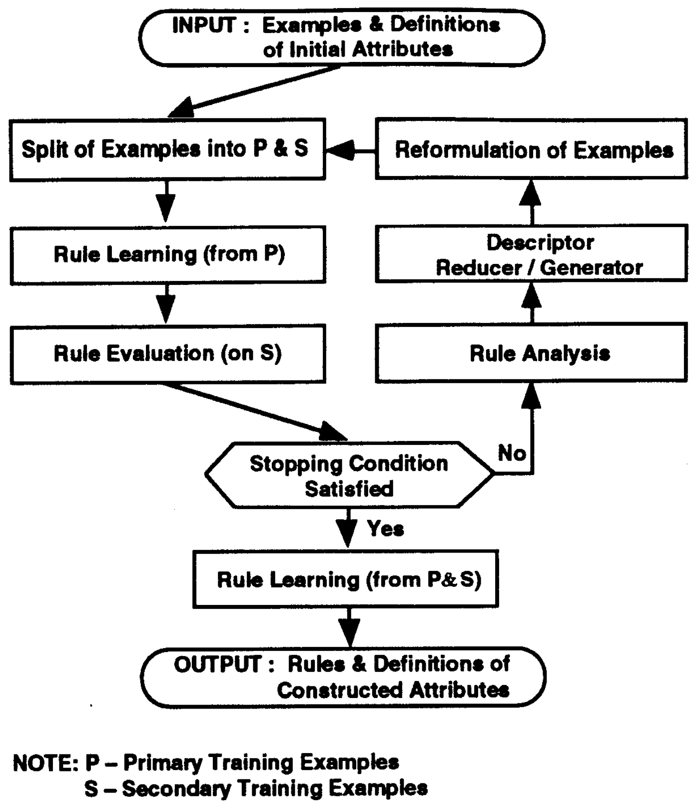


Figure 1. The HCI method for hypothesis-driven constructive induction.

The "Rule Learning" module induces a set of decision rules from the primary training set P, using the AQ15 inductive learning program (Michalski et al. 1986). The program employs the algorithm $A^q$ for solving general covering problem, which has been described in various sources, e.g., (Michalski 1983).

The "Rule Evaluation" module determines the performance of the rules on the secondary training set, S. The performance is measured in terms of predictive accuracy of the rules. The result is compared with the predefined threshold. If the performance accuracy does not satisfy the Stopping Condition, the rules are analyzed to determine desirable changes in the representation space. The analysis of the hypothesis in the "Rule Analysis" module determines which of the original attributes are redundant, and which rules perform best for each decision class (or concept).

In the "Descriptor Reducer/Generator" module, the redundant attributes are removed from the representation space. The best-performing rules are assembled into sets that are assigned names, and treated as new attributes that extend the representation space. Training examples are projected into the new representation space ("Reformulation of Examples" module), and the inductive process is repeated.

If the performance accuracy exceeds a predefined threshold, or chances for further improvement are estimated as being below a certain limit, final induction of decision rules is performed in the "Rule Learning" module from the complete training set.

## DEVELOPMENT OF EXAMPLES

Decision rules were learned from a collection of 336 examples of minimum weight (optimal) designs of wind bracings in steel skeleton structures of tall buildings. Each example has three major components: 1) a description of the building for which a given bracing is intended (design case or design requirements), 2) a description of the wind bracing structural system, and 3) an evaluation of the unit steel weight of a given wind bracing for the design case considered. Therefore, each example relates the design requirements to the selection of components of a wind bracing structural system and the unit steel weight of this system. Machine learning is used to produce decision rules which explain how design requirements can be optimally satisfied through the proper selection of individual components of a wind bracing structural system. This is an extremely complex structural design problem, and any progress toward its solution is important.

The application area for our study has been selected for significant technical reasons. At present, the analysis, design, and optimization of wind bracings in steel skeleton structures can be conducted using computer packages such as SODA.* The use of these packages makes the design of wind bracings easier, but it still does not eliminate the need for the proper selection of a wind bracing type for a given design case. This selection must be based, as before, on the designer's experience. A computer package can produce a feasible locally optimal design of any type of wind bracing; however, this may not be the global optimum. For example, a computer package may produce a minimum-weight design for wind bracing in the form of a one-bay rigid frame in a twenty-story skeleton structure. This design may not be a global optimal design: much better weight and stiffness characteristics might be obtained with a truss wind bracing. The problem of selecting the proper type of wind bracing is particularly important when inexperienced designers use design and optimization computer packages; their lack of experience may lead to feasible designs which could be significantly improved through simple changes in configuration.

For these reasons, research at Wayne State University on automated knowledge acquisition of wind bracings selection for steel skeleton structures was initiated in cooperation with Donald Grierson of Waterloo University, Ottawa, Canada. The knowledge acquired will eventually be used in a knowledge-based system which will be combined with SODA. This knowledge-based system will guide SODA users in the process of selecting the most appropriate types of wind bracings for their individual design cases. The research was initiated in 1986 by Arciszewski and Mustafa (Arciszewski and Ziarko 1987, Mustafa and Arciszewski 1989, Mustafa and Arciszewski 1992) in the Intelligent Computers Laboratory of the Civil Engineering Department at Wayne State University, Detroit.

All examples were prepared under identical design assumptions for a three-bay skeleton structure of a tall building. An eight-attribute description of the design problem and the wind bracing itself was developed for this case (Mustafa 1989). The attributes used are sufficient to describe various types of flat frame, truss, and truss-frame bracings which are appropriate for buildings in the six to thirty-story height range considered in our study. These attributes are based on a general description of wind bracings in tall buildings proposed by Arciszewski (1985) and used by Mustafa and Arciszewski (1992) in earlier machine learning experiments in the area of knowledge acquisition of conceptual wind bracing design. The individual attributes and their values are given in Table 4.

The first three attributes, *Number of Stories*, *Bay Length*, and *Importance Factor*, describe the design case (design requirements) considered. Attributes No. 4 through No. 7 describe the structural system of wind bracing itself, while the last attribute, No. 8, *Unit Steel Weight*,

* SODA is a Structural Optimization, Design and Analysis package for planar steel frames and trusses under static load. It was developed by Waterloo Engineering Software, Ontario, Canada.

identifies the nominal value of the relative unit weight of the steel structural system of a wind bracing described by attributes one through seven. This relative unit steel weight is determined considering all normalized unit weights of various types of wind bracings of the same height designed under identical conditions.

Examples were prepared by Mohamad Mustafa (1989) as part of his doctoral research on an engineering methodology of automated knowledge acquisition. All detailed design assumptions regarding loads, dimensions, steel grade, etc., were determined in cooperation with practicing structural designers. Actual minimum-weight designs were produced using SODA. All designs were verified by Mustafa, who is a structural designer with the Wayne County Structural Division.

Table 4. Attributes and Their Values.

| ATTRIBUTES | ATTRIBUTE VALUES | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1. Number of Stories | 6 | 12 | 18 | 24 | 30 |
| 2. Bay Length | 20 | 30 | | | |
| 3. Importance Factor | 1.07 | 1.11 | | | |
| 4. Joints | Rigid | Hinged | Mixed | | |
| 5. Number of Bays | 1 | 2 | 3 | | |
| 6. Number of Vertical Trusses | 0 | 1 | 2 | 3 | |
| 7. Number of Horizontal Trusses | 0 | 1 | 2 | 3 | |
| 8. Unit Steel Weight | Low | Medium | High | Infeasible | |

The decision rules obtained are divided into four classes corresponding to the values of the decision attribute *Unit Steel Weight.* Values of the unit weight are specified in the examples as low, medium, high, and infeasible when SODA could not produce a wind bracing under the given assumptions which would satisfy all design requirements. Therefore, the decision rules which specify designs with low unit weights are called *Recommendation Rules*. Similarly, decision rules which produce medium unit weights are called *Standard Rules*, and rules which produce high unit weight are called *Avoidance Rules*. Rules producing infeasible wind bracings are called *Infeasibility Rules.*

## LEARNING DECISION RULES

### Specific Knowledge Acquisition Process

In earlier research on machine learning about design of wind bracings in tall buildings (Mustafa and Arciszewski 1992), a collection of examples of various designs was prepared by human experts. The results of learning were found to be only partially satisfactory. Not all decision rules were correct from the structural point of view, and their interpretation was difficult. The problem is believed to be due mostly to an insufficient number of examples (165) and to the possible misclassification of some examples. The complexity of problem prevented the preparation of more examples and the improvement of the existing ones. Therefore, it was decided that a knowledge acquisition process be developed, based on the use of computer-generated examples of optimal wind bracings designs. In this way, a larger number of examples could be produced and their quality monitored. The process developed is shown in Fig. 2. In this process, SODA is used to generate examples, and two experimental learning systems, based on data-driven and hypothesis-driven constructive induction, are used to induce decision rules.

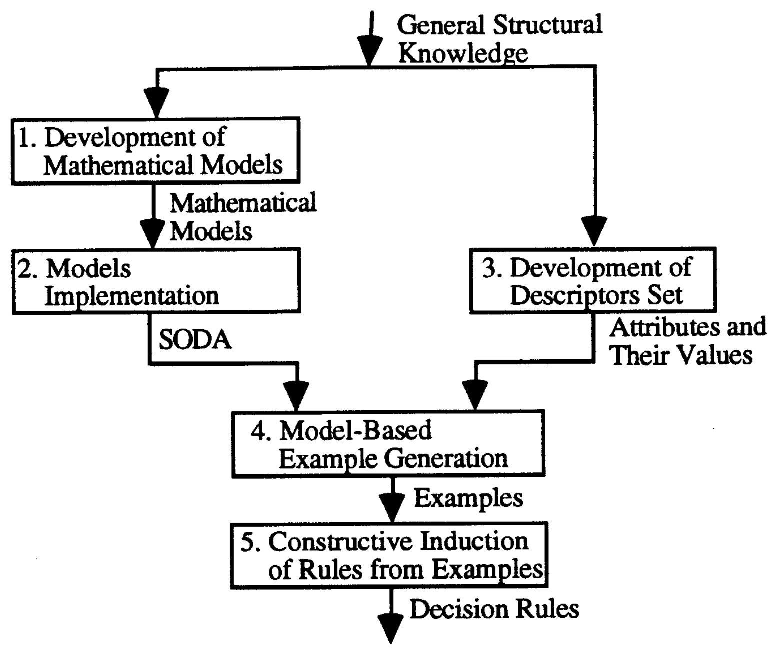


Figure 2. Process of Knowledge Acquisition.

Five major stages are distinguished in the proposed knowledge acquisition process:

1. **Development of Mathematical Models.** In this stage, the available general structural knowledge was used by Donald Grierson to develop a system of mathematical models for the analysis, design, and optimization of steel structures, which could be consistently implemented in computer programs. Modeling was initiated in the early sixties at Waterloo University, and the majority of work was completed in the late eighties (Grierson 1989, Grierson and Cameron 1989).

2. **Implementation of Models.** This stage involved the development of a computer package, SODA, for the analysis, design, and optimization of steel structures. The implementation was initiated in the mid-eighties by Donald Grierson at Waterloo Engineering Software, and is still under development.

3. **Development of Descriptor Set**. The work included the identification of relevant nominal attributes describing wind bracings in steel skeleton structures and the determination of their values. The work was initiated in the early seventies by Tomasz Arciszewski at Warsaw Technical University and continued at Wayne State University in Detroit (Arciszewski 1985).

4. **Model-Based Example Generation.** This stage involved the preparation of a collection of 336 examples of SODA-generated minimum-weight wind bracing designs of various types. It was initiated in 1989 by Mohamad Mustafa and Tomasz Arciszewski at Wayne State University, and took approximately two years to complete.

5. **Constructive Induction of Decision Rules from Examples.** In this stage, two experimental learning systems, based on data-driven and hypothesis-driven constructive induction, were used to produce decision rules. The work was conducted at the Center for Artificial Intelligence at George Mason University in 1992, and it took several weeks to complete induction and to interpret results.

Stage one in the knowledge acquisition process was concerned the development of deep models. In stage two, a simulation model was produced. In stage three, relevant attributes were proposed. Stage four, Model-Based Example Generation, can be considered as knowledge deduction from the deep model through the use of the simulation model. The last stage, Constructive Induction of Rules from Examples, can be interpreted as knowledge compression.

Two knowledge induction processes were conducted:
1. Generation of decision rules from examples using data-driven constructive induction, and
2. Generation of decision rules from examples using hypothesis-driven constructive induction.

### Data-Driven Constructive Induction of Decision Rules

The learning system was provided with several relationships among various combinations of attributes. These relationships were then used by the learning system to produce decision rules.* Individual collections of decision rules were analyzed later by a domain expert from the point of view of their domain relevancy and clarity. One relationship, in the form of the product of attributes X1 and X3 divided by X2, produced a particularly interesting and consistent collection of decision rules. This new relationship is a constructed attribute called *Demand,* and is defined as:

$$Demand = \frac{X1 * X3}{X2}$$

The new attribute Demand can be interpreted as a measure of demand for bracings: it increases with the height of the building (measured by attribute No. 1, Number of Stories) and the importance factor which is associated with the wind exposure (represented by attribute No. 3, Importance Factor) and decreases with the width of the building (measured by attribute No. 2, Bay Length). In the collection of examples considered, the new attribute Demand has the domain (0.214 .. 1.665). This domain was divided into several subranges, each with a clear technical interpretation:

1. Subrange 0.214 .. 0.500 – very low demand
2. Subrange 0.500 .. 0.780 – low demand
3. Subrange 0.780 .. 1.060 – average demand
4. Subrange 1.060 .. 1.340 – high demand
5. Subrange 1.340 .. 1.665 – very high demand

Six *Recommendation Rules*, thirteen *Standard Rules*, ten *Avoidance Rules*, and one *Infeasibility Rule* have been obtained. All these decision rules, with the exception of *Standard Rules*, are presented and their domain interpretation is provided. *Standard Rules* are correct, but their number and complexity exclude the possibility of discussion here. Decision rules are shown in descriptive form and in parentheses as they were produced by the learning system. Interpretation is also provided.

* In this case the DCI method was used as a tool for quickly evaluating the usefulness of supplied combinations rather than in the discovery of new attributes. Work toward providing DCI with sufficient domain knowledge to discover relevant attributes in such complex areas as design is underway.

The following *Avoidance Rules* (AR) were produced:

```
AR1:   IF    (Number of Bays IS 1 or 2) &
             (Number of Vertical Trusses IS 0) &
             (Demand IS very low)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is very low, then avoid designing a wind bracing in the form of a single or double one-bay rigid frame without vertical trusses.

```
AR2:   IF    (Joints ARE rigid or mixed) &
             (Number of Horizontal Trusses IS 0 or 2) &
             (Demand IS very low)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is very low, then avoid designing a wind bracing in the form of a rigid frame, or in the form of a rigid frame with two horizontal trusses.

```
AR3:   IF    (Joints ARE mixed) &
             (Number of Bays IS 1 or 2) &
             (Demand IS very low)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is very low, then avoid designing a wind bracing in the form of a one-bay bracing with mixed joints (braced rigid frame) or in the form of two braced rigid frames.

```
AR4:   IF    (Number of Bays IS 1) &
             (Number of Vertical Trusses IS 0) &
             (Number of Horizontal Trusses IS 1) &
             (Demand IS very low)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is very low, then avoid designing a wind bracing in the form of a one-bay rigid frame with one horizontal truss.

```
AR5:   IF    (Number of Bays IS 1) &
             (Number of Vertical Trusses IS 0) &
             (Demand IS average)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is average, then avoid designing a wind bracing in the form of a single bay rigid frame.

```
AR6:   IF    (Importance Factor IS high) &
             (Joints ARE rigid) &
             (Demand IS average)
       THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is average and the building is located in a high wind intensity zone, then avoid designing wind bracings with rigid joints only without any trusses.

```
AR7:   IF    (Number of Stories IS 12) &
             (Bay Length IS 20) &
             (Joints ARE mixed) &
             (Number of Bays IS 2) &
             (Number of Horizontal Trusses IS 3)
       THEN  a high unit weight of bracing should be expected.
```

Avoid designing two-bay rigid frames with three horizontal trusses for narrow twelve-story buildings.

```
AR8:    IF    (Number of Stories  IS  30)  &
              (Joints  ARE  mixed)  &
              (Number of Bays  IS  2)  &
              (Number of Horizontal Trusses  IS  1 or 2)
        THEN  a high unit weight of bracing should be expected.
```

Avoid designing a thirty story building with wind bracings in the form of two-bay rigid frames with one or two horizontal trusses.

```
AR9:    IF    (Joints  ARE  mixed)  &
              (Number of Bays  IS  1)  &
              (Number of Horizontal Trusses  IS  1)  &
              (Demand  IS  average)
        THEN  a high unit weight of bracing should be expected.
```

If demand for bracing is average, then avoid designing wind bracings in the form of a single-bay rigid frame with one horizontal truss.

```
AR10:   IF    (Number of Stories  IS  24)  &
              (Number of Bays  IS  3)  &
              (Number of Horizontal Trusses  IS  3)
        THEN  a high unit weight of bracing should be expected.
```

Avoid designing a 24-story building with three-bay wind bracing with three horizontal trusses (a three-bay rigid frame with three horizontal trusses).

The following seven *Recommendation Rules* (RR) were obtained:

```
RR1:    IF    (Number of Vertical Trusses  IS  1..3)  &
              (Demand  IS  very low or low or average)
        THEN  a low unit weight of bracing should be expected.
```

If demand is very low, low, or average, then design wind bracings with vertical trusses.

```
RR2:    IF    (Number of Stories  IS  30)  &
              (Number of Vertical Trusses  IS  1 or 2)  &
              (Number of Horizontal Trusses  IS  1..3)
        THEN  a low unit weight of bracing should be expected.
```

For thirty-story buildings, design wind bracing systems using combinations of vertical and horizontal trusses.

```
RR3:    IF    (Number of Bays  IS  2 or 3)  &
              (Number of Horizontal Trusses  IS  1..3)  &
              (Demand  IS  very low)
        THEN  a low unit weight of bracing should be expected.
```

If demand for bracing is very low, design two- or three-bay wind bracings with horizontal trusses.

```
RR4:     IF     (Number of Stories  IS  30)  &
                (Number of Vertical Trusses  IS  1)
         THEN   a low unit weight of bracing should be expected.
```

Design thirty-story buildings with wind bracings containing one vertical truss.

```
RR5:     IF     (Number of Vertical Trusses  IS  3)
         THEN   a low unit weight of bracing should be expected.
```

Design wind bracings with three vertical trusses.

```
RR6:     IF     (Number of Vertical Trusses  IS  3)  &
                (Number of Horizontal Trusses  IS  3)  &
                (Demand  IS  very low)
         THEN   a low unit weight of bracing should be expected.
```

When demand for bracing is very low, then design wind bracings with three vertical and three horizontal trusses. (This case would very rarely occur because wind bracing with three vertical trusses normally do not have any additional horizontal trusses.)

```
RR7:     IF     (Joints  ARE  rigid)  &
                (Number of Bays  IS  2)  &
                (Demand  IS  high)
         THEN   a low unit weight of bracing should be expected.
```

When demand for bracing is high, then design wind bracings in the form of two single one-bay rigid frames.

The last decision rule, RR7, is incorrect. It has been produced because of an error in the input data. Example No. 298 had an incorrectly entered value of the decision attribute - low unit weight instead of high unit weight. Therefore, a recommendation rule instead of an avoidance rule was produced. This decision rule is shown and discussed here to demonstrate how sensitive the learning system is sometimes to incorrect examples.

Only one *Infeasibility Rule* (IR) was produced:

```
IR1:     IF     (Joints  ARE  rigid or mixed)  &
                (Number of Bays  IS  1)  &
                (Demand  IS  high)
         THEN   wind bracing will be infeasible.
```

When demand for bracing is high, never design wind bracings in the form of a single one-bay structural system with rigid or mixed joints.

### Hypothesis-Driven Constructive Induction of Decision Rules

The learning system conducted a three-stage knowledge acquisition process. In the first stage, abstraction of the knowledge in the form of examples was performed and the representation space was reduced. Attribute No. 3, Importance Factor, was eliminated. This result is not entirely unexpected, because this attribute is rarely a decisive factor in the structural shaping of wind bracings. In the second stage of knowledge acquisition, the system conducted concretion of knowledge and constructed a new attribute with five values. This new attribute, called the Constructed Attribute, or CA, is defined as follows:

```
IF      (Number of Stories  IS  6)  &
        (Number of Bays  IS  1 or 2)  &
        (Number of Vertical Trusses  IS  0)  &
        (Number of Horizontal Trusses  IS  0..2)
         OR
        (Number of Stories  IS  6)  &
        (Bay Length  IS  30)  &
        (Number of Vertical Trusses  IS  0)  &
        (Number of Horizontal Trusses  IS  0..2)
         OR
        (Number of Stories  IS  6)  &
        (Bay Length  IS  30)  &
        (Number of Bays  IS  1 or 2)  &
        (Number of Vertical Trusses  IS  0)
THEN    CA = 1
```

```
IF      (Number of Stories  IS  12..30)  &
        (Number of Bays  IS  3)  &
        (Number of Vertical Trusses  IS  0)  &
        (Number of Horizontal Trusses  IS  0..2)
         OR
        (Number of Stories  IS  12..24)  &
        (Bay Length  IS  30)  &
        (Joints  ARE  rigid or mixed)  &
        (Number of Bays  IS  1 or 2)  &
        (Number of Horizontal Trusses  IS  0 or 2 or 3)
         OR
        (Number of Stories  IS  18 or 24)  &
        (Joints  ARE  mixed)  &
        (Number of Horizontal Trusses  IS  1 or 3)
THEN    CA = 2
```

```
IF      (Number of Stories  IS  6..24)  &
        (Number of Bays  IS  2 or 3)  &
        (Number of Vertical Trusses  IS  1..3)
         OR
        (Number of Stories  IS  12..30)  &
        (Joints  ARE  hinged)  &
        (Number of Horizontal Trusses  IS  1..3)
THEN    CA = 3
```

```
IF      (Number of Stories  IS  30)  &
        (Joints  ARE  rigid or mixed)  &
        (Number of Bays  IS  1)
THEN    CA = 4
```

```
IF      (none of the above formulas is satisfied)
         OR
        (more than one formula is satisfied)
THEN    CA = 5
```

All values of the Constructed Attribute have a clear structural engineering meaning and can be interpreted in the terms of structural shaping of wind bracings. For example, the value **1**, (CA = **1**) can be interpreted in the following way:

| | |
|---|---|
| *In order to obtain a low unit weight of bracing for a six-story building, avoid designing a wind bracing as a rigid frame with the following three combinations of structural attributes:* | (Number of Stories IS **6**) & (Number of Vertical Trusses IS **0**) |
| *(1) a single one-bay or two one-bay structural system with or without one or two horizontal trusses* | (Number of Bays IS **1 or 2**) & (No H. Trusses IS **0..2**) |
| *OR* | |
| *(2) with a wide bay and with or without one or two horizontal trusses* | (Bay Length IS **30**) & (No H. Trusses IS **0..2**) |
| *OR* | |
| *(3) with a wide bay and as a single one-bay or two one-bay structural system* | (Bay Length IS **30**) & (Number of Bays IS **1 or 2**) |

In the third stage of knowledge acquisition, the system produced four classes of decision rules, including five *Avoidance Rules*, five *Standard Rules*, three *Recommendation Rules*, and one *Infeasibility Rule*. All decision rules, with the exception of *Standard Rules*, are presented and their domain interpretation provided:

*Avoidance Rules* (AR):

```
AR1:      IF    (CA  IS  1)
          THEN  a high unit weight of bracing should be expected
```

The interpretation is given above as the explanation of the constructed attribute *CA*.

```
AR2:      IF    (Number of Bays  IS  1 or 3)  &
                (Number of Vertical Trusses  IS  0)  &
                (Number of Horizontal Trusses  IS  1)  &
                (CA  IS NOT  2)  &
                (CA  IS NOT  4)
          THEN  a high unit weight of bracing should be expected.
```

Avoid designing a one- or three-bay wind bracing in the form of a rigid frame, use one horizontal truss and make sure that the value of constructed attribute CA is neither two or four.

```
AR3:      IF    (Number of Stories  IS  12)  &
                (Bay Length  IS  20)  &
                (Joints  ARE  mixed)  &
                (Number of Bays  IS  2)  &
                (Number of Horizontal Trusses  IS  3)
          THEN  a high unit weight of bracing should be expected.
```

Avoid designing twelve-story buildings with a narrow bay with wind bracings in the form of two one-bay rigid frames and three horizontal trusses.

```
AR4:    IF     (Number of Stories  IS  18)  &
               (Number of Vertical Trusses  IS  0)  &
               (Number of Horizontal Trusses  IS  0)  &
               (CA  IS NOT  2)
        THEN   a high unit weight of bracing should be expected.
```

Avoid designing eighteen-story buildings with wind bracings in the form of rigid frames and make sure that the constructed attribute CA is not equal two.

```
AR5:    IF     (Number of Stories  IS  18)  &
               (Joints  ARE  mixed)  &
               (Number of Bays  IS  1)  &
               (CA  IS NOT  2)
        THEN   a high unit weight of bracing should be expected.
```

Avoid designing eighteen-story buildings with wind bracings in the form of a single one-bay rigid frame with horizontal truss or trusses and make sure that the constructed attribute CA is not equal two.

Three *Recommendation Rules* (RR) were produced:

```
RR1:    IF     (CA  IS  3)
        THEN   a low weight of bracing should be expected.
```

In designing wind bracings in tall buildings, select bracings which are described by a combination of attributes and their value which causes that the constructed attribute CA is equal 3.

```
RR2:    IF     (Bay Length  IS  20)  &
               (Joints  ARE  hinged)
        THEN   a low weight of bracing should be expected.
```

Use truss bracings for narrow bay buildings.

```
RR3:    IF     (Number of Stories  IS  12..30)  &
               (Bay Length  IS  20)  &
               (Number of Bays  IS  1 or 3)  &
               (Number of Horizontal Trusses  IS  0 or 3)  &
               (CA  IS NOT  2)  &
               (CA  IS NOT  4)
        THEN   a low weight of bracing should be expected.
```

Design wind bracings for twelve-,eighteen-, and thirty-story buildings with narrow bay in the form of one- or three-bay structural systems without or with three horizontal trusses and make sure that the value of constructed attribute CA is neither two nor four.

A single *Infeasibility Rule* (IR) was produced:

```
IR1:    IF     (CA  IS  4)
        THEN   the wind bracing will be infeasible.
```

Never design wind bracings if the constructed attribute CA is equal to four.

This is equivalent to the rule

```
IR1:     IF     (Number of Stories  IS  30)  &
                (Joints  ARE  rigid or mixed)  &
                (Number of Bays  IS  1)
         THEN   the wind bracing will be infeasible.
```

In the case of a thirty-story building, never design wind bracings in the form of a single one-bay structural system with rigid or mixed joints.

## PERFORMANCE ANALYSIS

One of this paper's objectives is to investigate the performance of constructive induction-based learning systems in the area of structural design knowledge acquisition. This performance can be formally measured by various empirical error rates, which are determined through tests. In each test, a learning system uses a given body of examples to make predictions about other known examples which have not been included in its input. Each test can be then compared to a real-life situation, when a designer uses a decision support system to predict the structural attributes of a wind bracing to minimize its weight. Therefore, empirical error rates are highly relevant to both machine learning research, which is concerned with the performance of learning systems, and to structural design, which is concerned with the optimal decision making.

Two empirical error rates were used: 1) the overall empirical error rate, and 2) the omission error rate. The overall empirical error rate was used because it provides the most general evaluation of performance of a learning system and the knowledge acquired. Also, it has a simple interpretation, convincing for structural designers. The omission empirical error rate is also important, because it measures the degree to which the learning system, using knowledge acquired, fails to recognize cases belonging to individual categories of the decision attribute.

The overall empirical error rate is defined (Arciszewski, Dybala and Wnek 1992) as:

$$E_{ov} = \frac{Number\ of\ errors}{Number\ of\ tests}$$

where: *error* is a misclassification of a testing example,
*Number of tests* is the number of classification tests.

The omission empirical error rate is defined (Arciszewski, Dybala and Wnek 1992) as:

$$E_{om} = \frac{\sum_{i=1}^{n} E_{om}^{i}}{n}$$

where: $n$ is the number of classes

$$E_{om}^{i} = \frac{Number\ of\ omission\ errors\ for\ class\ "i"}{Number\ of\ tested\ examples\ of\ class\ "i"}$$

*Number of omission errors for class "i"* is the number of errors when a positive example is classified as a negative one
*Number of tested examples of class "i"* is the number of classification tests for class "i"

Both error rates were calculated for the entire collection of examples using the leave-one-out resampling method (Weiss and Kulikowski 1991). These error rates were determined for "traditional" induction, based on the use of the AQ15 algorithm, and for two experimental learning systems based on data-driven and hypothesis-driven constructive induction. Individual error rates are shown in the Table 5. There is a significant improvement in performance (more than 50 percent) between the system based on "traditional induction" and systems based on constructive induction. The differences in performance between the two constructive induction-based systems are insignificant (less than 5 percent), but this may change as the research progresses.

Table 5. Comparison of Empirical Error Rates for Various Learning Systems.

| | | CONSTRUCTIVE INDUCTION | |
|---|---|---|---|
| | AQ15 | Data-driven | Hypothesis-driven |
| Overall Error Rate (%) | 4.5 | 3.3 | 3.3 |
| Omission Error Rate (%) | 4.2 | 3.2 | 3.1 |

## CONCLUSIONS

The results demonstrate that the use of constructive induction in structural design knowledge acquisition is feasible. The decision rules produced are relatively simple and their structural interpretation is possible, although not always easy, particularly when complex constructed attributes are used.

The changes in the representation space, the result of abstraction and concretion which are characteristic of constructive induction, are acceptable to human experts. These changes are similar to those associated with the growing human understanding of a given domain, and therefore could also stimulate the process of human learning, but much more research is necessary to determine how constructive induction could be used in human learning.

The five-stage knowledge acquisition process used in this research is complete and sufficient for practical purposes. However, it is applicable only to those structural design domains where mathematical and simulation models are available, or can be easily produced. The knowledge representation stage was found to be particularly difficult. It required extensive study and the cooperation of experts. The identification of relevant attributes and their nominal values began in the early seventies, and these attributes have undergone numerous changes and modifications before a final acceptable set was produced. Including mathematical modeling, simulation and knowledge representation, the entire process of knowledge acquisition took about thirty years, although its last two stages, Knowledge Deduction and Knowledge Induction, were completed in about two years.

The learning system based on data-driven constructive induction produced ten *Recommendation Rules*, thirteen *Standard Rules*, seven *Avoidance Rules*, and one *Infeasibility Rule*. The learning system based on hypothesis-driven constructive induction produced four *Recommendation Rules*, five *Standard Rules*, three *Avoidance Rules*, and one *Infeasibility Rule*, respectively. The rules produced by the two systems are different, but not inconsistent. An analysis of advantages and disadvantages of rules produced by individual systems is possible, but it has not been conducted because the results would be subjective. However, an objective analysis of performance of learning systems based on the two forms of constructive induction was completed.

The performance analysis shows that both forms of constructive induction are more effective in terms of empirical error rates than traditional selective induction based on the AQ15 learning algorithm. An approximately 50 percent performance improvement is considered significant in machine learning, and should be accepted as such in civil engineering. The differences in performance between two systems based on different forms of constructive induction are below 5 percent, and they are obviously insignificant. Therefore, both forms of constructive induction should be considered equivalent from the performance point of view in the area of structural design knowledge acquisition. It would be desirable to produce and analyze learning curves for empirical error rates for both systems, to learn more about their performance in a multistage automated knowledge acquisition process, and this work is planned.

The feasibility study was conducted in the area of structural design knowledge acquisition, and therefore all conclusions produced are valid only in this area. However, it could be inferred by analogy that similar results, in terms of clarity of decision rules and good performance, should be expected in other areas of civil engineering.

Machine learning research has already reached a fair level of maturity, and has resulted in various experimental and commercial learning systems. These systems could be used in civil engineering to improve productivity in knowledge acquisition and the development of knowledge-based decision support systems. However, the application of learning systems is currently delayed by the lack of a methodology of their use, and the development of this methodology is becoming crucial for further progress. This paper provides some initial methodological results, but much more needs to be done. Any work on the methodology of applying learning systems to civil engineering will be difficult, but this is a challenge which must be met.

## REFERENCES

Adeli, H. and Yeah, C., (1989). "Perception Learning in Engineering Design," *Microcomputers in Civil Engineering*, No. 4.

Arciszewski, T., Dybala, T. and Wnek, J., (1992). "Method for Evaluation of Learning Systems," *Journal of Knowledge Engineering "Heuristics,* Vol. 5, No. 4.

Arciszewski, T. and Dybala, T., (1992). "Evaluation of Learning Systems: A Method and Experimental Results," *Report,* Center for Artificial Intelligence, George Mason University.

Arciszewski, T. and Mustafa, M., (1989). "Inductive Learning Process: The User's Perspective," in *Machine Learning*, R. Forsyth (Ed.), Chapman and Hall, pp. 39-61.

Arciszewski T. and Ziarko W., (1987). "Adaptive Expert System for Preliminary Engineering Design," *Revue Internationale de CFAO et D'Intographie*, Vol. 2, No. 1.

Arciszewski, T., Mustafa, M. and Ziarko, W., (1987). "A Methodology of Design Knowledge Acquisition for Use in Learning Expert Systems," *Int. J. Man-Machine Studies*, Academic Press Limited, 27, pp. 23-32.

Arciszewski, T., (1985). "Decision Making Parameters and their Computer-Aided Analysis for Wind Bracings in Steel Skeleton Structures," *Advances in Tall Buildings*, Van Nostrand Publishing Company.

Bala, J., Michalski, R.S. and Wnek, J., (1992). "The Principal Axes Method for Noise Tolerant Constructive Induction," *Proceedings of the Ninth International Conference on Machine Learning Conference*, Scotland, pp. 20-29.

Garrett, J. H. and Ivezic, N., (in print). "A Neural Network Approach for Acquiring and Using Synthesis Knowledge," *Proceedings of the Japan-U.S. Workshop on Expert Systems and AI Applications in Civil and Structural Engineering*, 25-26 August, 1992, Tokyo.

Gero, J.S., Mackenzie, C.H., and McLaughlin, S., (1989). "Learning from Optimal Solutions to Design Problems," *Proceedings of NATO ASI Conference on Optimization and Decision Support Tools in Civil Engineering*, Edinburgh.

Grierson, D.E., (1989). "Computer-Automated Optimal Design for Structural Steel Frameworks," *Proc. of NATO ASI Conf. on Optimization and Decision Support Systems in Civil Engineering*, Edinburgh.

Grierson, D.E. and Cameron, G.E., (1989). "Microcomputers -Based Optimization of Steel Structures in Professional Practice," *Microcomputers in Civil Engineering*, Elsevier Publishing, Vol 4. pp. 289-296.

Grierson, D.E., Pak, W.H., (1992). "Discrete Optimal Design Using a Genetic Algorithm," *Proceedings of NATO Conference on Topology Design of Structures,* Portugal.

Hajela, P., (1989). "Genetic Algorithms in Automated Structural Synthesis," *Proc. of NATO ASI Conf. on Optimization and Decision Support Systems in Civil Engineering*, Edinburgh.

Lu, S.C-Y. and Chen, K., (1987). "A Machine Learning Approach to the Automatic Synthesis of Mechanistic Knowledge for Engineering Decision Making," *Journal Artificial Intelligence in Design and Manufacturing*, Vol. 1, No. 2.

Mackenzie, C.A. and Gero, J.S., (1987). "Learning Design Rules from Decisions and Performances," *Artificial Intelligence in Engineering*, Vol. 2, No. 1.

Maher, M.L., (1992a). "Automated Knowledge Acquisition of Preliminary Design Concepts," *Proceedings of the ASCE Conference for Computing in Civil Engineering*, Dallas.

Maher, M.L. and Li, H., (1992b). "Automatically Learning Preliminary Design Knowledge from Design Examples," *Microcomputers in Civil Engineering*, Vol. 7, No. 1.

McLaughlin, S. and Gero, J.S., (1987). "Acquiring Expert Knowledge from Characterized Designs," *AIEDAM*, Vol. 1, No. 2.

Michalski, R.S., (1978). "Pattern Recognition as Knowledge-Guided Computer Induction," *Report*, Computer Science Dept., University of Illinois, Urbana.

Michalski, R.S. (1983). "A Theory and Methodology of Inductive Learning," In *Machine Learning: An Artificial Intelligence Approach*, Vol. I, R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), Morgan Kaufmann, Los Altos, CA.

Michalski, R.S. and Stepp, R.E., (1986). "Conceptual Clustering of Structured Objects: A Goal-Oriented Approach," *Artificial Intelligence*, Vol. 28, pp. 43-69.

Michalski, R.S., (to appear 1993). "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," *Machine Learning: A Multistrategy Approach*, Vol. IV, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, San Mateo, CA. (An earlier version published in *Reports of Machine Learning and Inference Laboratory*, MLI 92-3, Center for AI, George Mason University, 1992).

Michalski, R.S. and Tecuci, G., (to appear 1993). *Machine Learning: A Multistrategy Approach*, Vol. IV, Morgan Kaufmann Publishers, San Mateo, CA.

Milzner, K. and Harbecke, A., (1992). "Incremental Learning for Improved Decision Support in Knowledge Based Design Systems," *Proceedings of the Second International Conference on Artificial Intelligence in Design 1992*, J.S. Gero (Ed.), Pittsburgh.

Mustafa, M., (1989). "Engineering Methodology of Automated Knowledge Acquisition: Structural Application," *Ph. D. Proposal*, Civil Engineering Department, Wayne State University, Detroit.

Mustafa, M. and Arciszewski T., (1992). "Inductive Learning of Wind Bracing Design for Tall Buildings," in *Knowledge Acquisition in Civil Engineering*, T. Arciszewski and L. Rossman (Eds.), the American Society of Civil Engineers.

Pawlak, Z., (1982). "Rough Sets," *Intl. Journal of Computer and Information Sciences*, No. 11, Vol. 5, pp. 341-356.

Reich, Y., (1991). "Design Knowledge Acquisition: Task Analysis and A Partial Implementation," *Knowledge Acquisition*, Vol. 3, No. 3, pp. 237-254.

Reich, Y., (1992). "Generation of Examples for Training a Learning Design System," *Proceedings of the ASCE Conference Computing in Civil Engineering*, Dallas.

Reich, Y. and Fenves, S., (1988). "Floor-System Design in SOAR: A Case Study of Learning to Learn," *Report*, Engineering Design Research Center, Carnegie Mellon University.

Reich, Y. and Fenves, S.J., (1992). "Automated Acquisition of Design Knowledge by Concept Formation," in *ASCE monograph Knowledge Acquisition in Civil Engineering*, T. Arciszewski and L. Rossman (Eds.).

Weiss, S.M. and Kulikowski, C.A., (1991). *Computers that Learn*, Morgan Kaufman Publishers, San Mateo, California.

Whitehall, B.L., Stepp, R.E. and Lu, S.C-Y, (1990). "Knowledge-Based Learning: Using Domain Theory to Guide Induction," *Annual Report of Knowledge-Based Engineering Systems Research Laboratory*, University of Illinois at Urbana-Champaign.

Winston, P.H., (1982). "Learning New Principles from Precedents and Exercises," *Artificial Intelligence*, Vol. 19.

Wnek, J. and Michalski, R.S., (1991). "Hypothesis-driven Constructive Induction in AQ17: A Method and Experiments," *Proceedings of the IJCAI-91 Workshop on Evaluating and Changing Representations*, K. Morik, F. Bergadano and W. Buntine (Eds.), pp. 13-22, Sydney, Australia. (An extended version will appear in the Special Issue on Representation, *Machine Learning,* 1993.)

Wnek, J. and Michalski, R.S., (to appear). "Comparing Symbolic and Subsymbolic Learning: Three Studies," in *Machine Learning: A Multistrategy Approach*, Vol. IV, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann Publishers, San Mateo, CA.

Ziarko, W., (1989). "Data Analysis and Case-Based Expert System Development Tool ROUGH," *Proceedings of the Workshop on Case-Based Reasoning*, Pensacola Beach, Florida, Morgan Kaufman Publishers.