Proceedings of the Third International Round-Table Conference on Computational Models of Creative Design, Heron Island, Queensland, Australia, December 3-7, 1995.

Reports of Machine Learning and Inference Laboratory, MLI 95-6, George Mason University, April 1995.

CONSTRUCTIVE INDUCTION: THE KEY TO DESIGN CREATIVITY

TOMASZ ARCISZEWSKI[1], RYSZARD S. MICHALSKI[2] AND JANUSZ WNEK
Machine Learning and Inference Laboratory, George Mason University
4400 University Dr., Fairfax VA 22030, USA
[1]Also with the GMU Systems Engineering Department
[2]Also with the GMU Departments of Computer Science and Systems Eng., and the Institute of Computer Science at the Polish Academy of Sciences

Abstract. The paper presents initial results from an emerging new direction in engineering design research, in particular, creative design. It argues that *constructive induction*, which was originally proposed in the field of machine learning, can serve as a foundation for developing a computational theory of engineering design and design creativity. Constructive induction is a process of creating new knowledge (e.g., design knowledge) by performing two intertwined searches, one—for the most adequate knowledge representation space, and second—for the best hypothesis in this space. Basic concepts and methods of constructive induction are reviewed and illustrated by examples of their application to conceptual structural design. Several crucial design concepts, including those of an emergent concept and of a goal-oriented transformation of the design representation space are interpreted in terms of a construction induction process. It is also shown how constructive induction applies to the control of the design creativity level. Several measures of the design complexity and relative creativity are proposed. The conclusion presents some unresolved problems and a plan for future research.

1. Introduction

Engineering is presently undergoing a paradigm change. The previous *analytical paradigm* is being gradually replaced by *a knowledge paradigm*, and this change appears to have a significant impact on the understanding of engineering design creativity. When the first paradigm was dominant, the major focus in engineering was to build analytical models of engineering systems in order to develop understanding of their behavior and to produce knowledge about them. In this context, engineering knowledge was understood as a combination of a representation of the engineering system being designed, of the design process itself, as well as of all relationships existing among attributes describing the system and their groups in the representation space.

At present, sufficient knowledge about engineering systems is frequently available; therefore, the focus in design is on how to utilize the available knowledge by means of information technology in the development of new designs. In this context, information technology signifies a cluster of related disciplines concerned with acquisition, processing, distribution and/or generation of information, or knowledge, using computer technology. These disciplines include computer science, artificial intelligence, machine learning, automated reasoning, decision science, software engineering, systems engineering, and others. The process of entering the knowledge paradigm can be compared to moving from medieval times to renaissance. One of the hallmarks of renaissance was Leonardo daVinci's observation that human artistic creativity, a spontaneous process of employing imagination and thinking to create art, can be used to produce engineering inventions and to solve engineering problems. In the knowledge paradigm, reasoning, invention and creativity are viewed as knowledge processing activities, and could be, at least partially, performed on a computer. Such a view is supported by the development of machine learning methods and their application to an automated knowledge creation and improvement. It is believed these efforts will likely lead to the development of computational foundations of conceptual design and to building a new class of design support tools. Such tools could, in turn, result in a new generation of inventions.

The above prospect creates an outstanding challenge for researchers and calls for study of design processes in terms of ideas developed in the area of artificial intelligence, particularly in machine learning and inference. Research in this new direction is at an early stage and, naturally, results are preliminary and have not made any significant impact on the design practice. This paper presents recent results of work on engineering design and design creativity in the context of constructive induction and the recently proposed Inferential Theory of Learning (Michalski, 1994). The material presented here rests on three ideas.

- The first idea is that the problems of design creativity can be usefully discussed by employing concepts and methods developed in the field of inductive learning. Any design process can be viewed as a search for a knowledge structure (a design, in this case) that satisfies given objectives and constraints. Such a process is a form of learning, therefore, ideas and methods developed in the field of machine learning may potentially bring new insights to the understanding of design processes.

- The second idea is that traditional learning methods, in which the search for desired knowledge occurs in the same representation space in which the original data are presented, are inadequate for understanding design processes. A more adequate approach is based on viewing the design processes as a form of *constructive induction.*

Constructive induction is a concept proposed in the field of inductive concept learning (Michalski, 1978a) to cope with learning problems in which the original representation space is inadequate for the problem at hand, and needs to be improved in order to correctly formulate the knowledge to be learned. More specifically, constructive induction is a process of hypothesizing new knowledge that involves not one search, as traditionally done, but two interrelated searches. The first search is for the "best" representation space in which desirable knowledge (e.g., a design) is to be searched for and represented. The second search is for the "best" hypothesis in the representation space which has been found. The underlying principle for this approach is that the desirable knowledge is easier to determine if the search for it is in the "right" representation space. It is claimed that partitioning a design process into two such searches can lead to novel and powerful models of design creativity. Thus, constructive induction can be viewed as the key to design creativity.

- The third idea is that the two searches in constructive induction can be conducted by applying *design knowledge transmutations*, as proposed in the Inferential Theory of Design (Arciszewski and Michalski, 1994). Design knowledge transmutations are generic types of design knowledge changes or generations, and have been based on more general knowledge transmutations, as first proposed in the Inferential Theory of Learning (Michalski, 1994).

The following sections present a more detailed exposition of the above ideas, discuss their significance for engineering design, and illustrate them by specific examples.

2. Basic Concepts and Assumptions

To explain the proposed view of design creativity as a form of constructive induction, we start with presenting basic concepts and assumptions.

ENGINEERING DESIGN

An engineering design is a description of an engineering system (that usually does not yet exist) expressed in terms of attributes defining a *representation space*. It consists of two major components: a *design concept* and a *detailed description*.

REPRESENTATION SPACE FOR AN ENGINEERING DESIGN

This is a multidimensional space spanned over attributes (in general, descriptive terms or descriptors) that are used to describe an engineering design (that is, a design concept and a detailed design description). Attributes can be symbolic (when they take values from an unordered or partially ordered set) or numerical (when they take numerical values representing quantities or measurements). Symbolic attributes that take values from an unordered set are called nominal attributes; when they take values from a partially ordered set, they are called structured. Design concepts are typically described in terms of symbolic attributes. Numerical attributes are used for a detailed description of a design.

DESIGN CONCEPT

A design concept describes a future engineering system in terms of abstract concepts, called "*primary concepts*," that involve nominal attributes and possibly also relations among design components. For example: the concept of a steel truss can be understood as a structural system whose description employs at least three primary concepts (symbolic attributes) and their values: the type of material with the value "steel," the member shape with the value "straight," and the type of connection with the value "pinned." Similarly, a concept of a belt truss system in wind bracings of a tall building employs primary concepts of a truss, a vertical truss, a truss grid, etc.

DETAILED DESCRIPTION

A detailed description of a design specifies values of all numerical attributes characterizing the design, such as specific dimensions, number of members, weight, etc.

DESIGNS COVERED BY A DESIGN CONCEPT

A given design concept represents a large class of specific engineering designs that differ in their detailed descriptions, i.e., in the values of numerical attributes characterizing these descriptions.

CONCEPTUAL DESIGN PARADIGMS

Five major conceptual design paradigms can be distinguished. This classification is based on the taxonomy proposed by Altschuller (1969), but has been modified and adapted in the context of the Inferential Design Theory:

1. Selection: the design concept is produced by selecting it from a class of known concepts in a given engineering domain.

2. Modification: the design concept is produced as a combination and/or modification of known design concepts from a given domain. The modification process is based on a deterministic or random generation process.

3. Innovation: the design concept is produced as a combination of known concepts from a given domain and other domains.

4. Invention: the design concept is produced as a combination of known concepts from a given domain and new concepts based on a new technology, which have been recently introduced.

5. Discovery: the design concept is produced as a combination of known concepts from a given domain and new concepts based on new scientific principles.

ROUTINE DESIGN

Routine design is a design process based only on selection or on modification. In both cases, no changes in the representation space occur.

CREATIVE DESIGN

Non-routine or creative design is a conceptual design process which is based on innovation, invention, or discovery. In all these cases, changes in a representation space occur. Thus, design processes are divided into creative or routine depending on whether the changes in the representation space occur or not, respectively.

ROUTINE VERSUS CREATIVE DESIGN

In general, there are two major differences between the routine and creative design processes: the number of changes of the representation space and the nature of inference. There are no changes in the representation space for routine design, and at least one change for non-routine or creative design. Routine design typically employs deductive inference (selection and modification), while creative design employs inductive inference (innovation, invention and/or discovery). For a definition and classification of inference types that are adopted in this paper, see (Michalski, 1994).

OPERATORS FOR REPRESENTATION SPACE TRANSFORMATION

Operators that change the representation space during the design process can be divided into four classes: attribute elimination (removing unimportant attributes), attribute abstraction (combining attribute values into larger units), attribute addition (adding new attributes to the representation space), and attribute construction (creating new attributes, called "*constructed attributes*" by a simple or complex transformation of the initial attributes). The last class is particularly interesting from the viewpoint of design creativity.

CONSTRUCTIVE INDUCTION

Constructive induction (in the context of engineering design) is a process of creating new knowledge (design) by two intertwined processes: one that searches for the best representation space for the design, and the second that searches for the best design in that space. Searching for the best representation space is done by applying *constructive induction operators* for the representation space transformation. This can be accomplished with the help of a human designer or automatically. In the first case, a human designer considers various design aspects (attributes from different representation spaces or primary concepts), intuitively conducting search for the best representation and looking for *emergent concepts* (constructed attributes) which might lead to a creative design. In the automated representation space transformation, constructive induction operators are applied by a computer program according to a constructive induction algorithm. Such an algorithm may use the advise of a designer as to the desirable representation space transformations and/or a set of predefined rules and methods. Constructive induction thus offers a formal methodology for characterizing and/or modeling a creative design process.

INDUCTIVE NATURE OF CREATIVE DESIGN

A creative design process is intrinsically inductive. It creates knowledge (design) that cannot be deductively derived from the original knowledge. The design is typically represented in a new representation space that was not initially given. A new representation space can be produced by employing knowledge from previous designs in other domains or by another process. In the former case, we say that constructive induction is guided by knowledge drawn from other domains. The new representation space may employ constructed attributes that appear to have high relevance for the design (*emergent concepts*). Their importance is recognized by human designers on the basis of their subjective understanding (background knowledge), and/or by applying methods for multicriterion evaluation. The evaluation criteria are usually acquired as the result of a human learning process about the problems being solved in the context of the designer's background knowledge.

EMERGENT CONCEPT

The emergent concept is a new concept created in the process of constructive induction that is potentially significant to the desired design. It provides a new understanding of the problem being solved and/or a new insight that is crucial for creative problem solving. The emergent concept is *a constructed attribute* whose introduction may simplify or improve a problem solving process (design process).

CONSTRUCTED ATTRIBUTE

A constructed attribute is derived from the initial attributes by an application of constructive induction operators (that can potentially involve any type of operation on attributes). Such an attribute has the potential to improve the effectiveness or quality of the problem solving process. This potential is estimated by some attribute evaluation function. A constructed attribute is usually a more abstract concept that the attributes/concepts from which it is derived, and it is supposed to improve the problem solving process and/or its understanding.

DESIGN KNOWLEDGE TRANSMUTATIONS

Design knowledge transmutations are knowledge operators that transform or generate new knowledge from other knowledge. They may use any form of inference in such a process. The Inferential Theory of Design introduced a class of 22 basic design knowledge transmutations or transforms (Arciszewski and Michalski, 1994; Michalski, 1994). These transmutations are considered to be major knowledge operators in conceptual design processes.

IMPLEMENTABILITY OF CONSTRUCTIVE INDUCTION

Knowledge space transformations employed in constructive induction are related to basic knowledge transmutations as defined in the Inferential Theory of Learning. For example, a contraction of the representation space (e.g., by removing attributes or abstracting attribute values) performs an *abstraction transmutation* of objects represented in this space. An *expansion* of the representation space (by adding new attributes or defining more specific values of attributes) performs a *concretion transmutation* of objects represented in this space. Computational aspects of constructive induction in engineering design go beyond the scope of this paper. We will only mention that design knowledge transmutations discussed here can be defined in a formal way and are therefore implementable.

POTENTIAL IMPACT OF MACHINE LEARNING ON DESIGN

Both learning and conceptual design processes are based on performing various forms of inference. Therefore, all experience from machine learning research that studies learning as an inferential process is relevant to design and can be used for developing a formal model of design processes. The initial formulation of the Inferential Design Theory is a result of such efforts.

3. Constructive Induction

3.1 BASIC CONCEPTS

Constructive induction (CI) employs a number of innovative ideas and assumptions:

- It is based on the idea that the quality of the knowledge representation space is the most important factor in concept learning. If the representation space is of high quality (i.e., chosen attributes or descriptive terms are of high relevance to the problem at hand), learning process will be relatively easy and will likely produce hypotheses with high predictive accuracy. If the quality of representation space is low (e.g., attributes are of little relevance to the problem), a learning process will be complex and no method may be able to produce good hypotheses.
- It searches for patterns in data and/or learned hypotheses, and uses them for proposing knowledge space transformations (that may expand or/and contract the space).
- It creates new descriptors (attributes or terms) that may be very complex, multilevel functions or transformations of the original descriptors.
- It postulates that produced concept descriptions should be comprehensible to human experts, so that they are relatively easy to interpret and express in terms and forms used by experts.

As mentioned earlier, constructive induction divides the process of creating new knowledge into a phase that determines the “best” knowledge representation, and a phase that actually formulates the desirable knowledge structure. The reason for such a division is that the original representation space, in which design cases, constraints, etc. are presented, is often inadequate for representing the sought design.

To illustrate this problem, consider Fig. 1A. Let us suppose that the problem is to construct a description that separates points marked by “ + ” from points marked by “-”. In this case, the problem is easy because “ + ” points can be separated from “-” points by a straight line or a rectangular border.

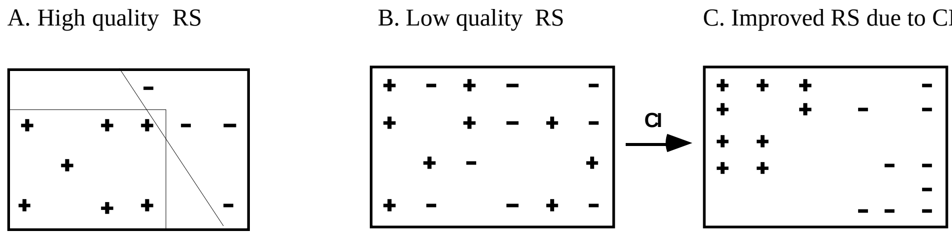


Figure 1. High vs. low quality representation spaces for concept learning.

Let us suppose now that "+"s and "-"s are distributed as in Fig. 1B. In this case, "+"s and "-" are highly intermixed, which may be an indication that the representation space is inadequate for the problem at hand. A traditional approach is to draw complex boundaries that will separate these two groups. The constructive induction approach is to search for a better representation space, such as shown in Figure 1C, in which the two groups are well separated.

Conducting constructive induction thus requires mechanisms for generating new, more problem-relevant dimensions of the knowledge representation space (attributes or descriptive terms) as well as modifying or removing less relevant dimensions from among those initially provided. In other words, a constructive induction system performs a problem-oriented transformation of the knowledge representation space. Once an appropriate representation space is found, a relatively simple learning method may suffice to develop a desirable knowledge structure (in this case, a description that separates the two groups of points).

3.2 CLASSIFICATION OF CONSTRUCTIVE INDUCTION APPROACHES

Research on constructive induction has produced a great variety of methods for that purpose (e.g., Larson and Michalski, 1977; Lenat, 1977; Langley et al., 1983; Utgoff, 1984; Rendell, 1985; Kokar, 1985; Flann and Dietterich, 1986; Schlimmer, 1987; Bentrup et al., 1987; Muggleton and Buntine, 1988; De Raedt and Bruynooghe, 1989; Drastal, Czako and Raatz, 1989; Matheus, 1989; Morik, 1989; Pagallo and Haussler, 1989; Knoblock, 1990; Wnek and Michalski, 1991, 1994a). A simple way to characterize these methods is to classify them in terms of the primary strategy employed for inventing new dimensions in the representation space (or generally *descriptors:* attributes, terms, or transformations) in order to improve the original representation space (a "representational bias"). Based on such criterion, constructive induction methods can be classified into the following categories:

Data–Driven Constructive Induction (DCI)
The representation space transformations are proposed on the basis of an analysis of the input data (examples) and a detection the interrelationships among descriptors. On that basis changes in the representation space are made. Examples of systems implementing DCI methods include: PLS0 (Rendell, 1985), AQ-DCI (Bloedorn and Michalski, 1991), BACON (Langley et al., 1983, 1987), ABACUS (Falkenhainer and Michalski, 1990; Greene 1988; Michael, 1991), Wyl, IOE (Flann and Dietterich, 1986; Flann 1990), STAGGER (Schlimmer, 1987), FCE (Carpineto, 1992).

Hypothesis–Driven Constructive Induction (HCI)
The representation space transformations are proposed on the basis of an analysis of inductive hypotheses generated in consecutive iterations. Patterns detected in one iteration are used in the next iteration. Examples of systems implementing HCI methods include: BLIP (Emde et al., 1983; Morik, 1989; Wrobel, 1989), FRINGE (Pagallo and Haussler, 1990), CITRE (Matheus, 1989), AQ-HCI (Wnek and Michalski, 1991, 1994a).

Knowledge–Driven Constructive Induction (KCI)
Expert-provided domain knowledge is used to construct a new representation space. Examples of systems implementing KCI methods include: INDUCE (Larson and Michalski, 1977), AM (Lenat, 1977, 1983); SPARC/E (Dietterich and Michalski, 1983, 1985, 1986), MIRO (Drastal et al., 1989), COPER (Kokar, 1986).

Multistrategy Constructive Induction (MCI)
Two or more strategies are combined for constructing a new representation space. Such methods include: INDUCE-1 (Larson and Michalski 1977; Michalski, 1978a, 1983), STABB (Mitchell et al., 1983; Utgoff, 1984, 1986), DUCE (Muggleton, 1987), CIGOL (Muggleton and Buntine, 1988), ALPINE (Knoblock et al., 1990,1991); CLINT (De Raedt and Bruynooghe, 1989).

The AQ-DCI and AQ-HCI methods have been developed in the course of our recent research (Wnek and Michalski, 1991, 1994ab; Wnek, 1993; Bloedorn and Michalski, 1991) in the Machine Learning and Inference Laboratory at George Mason University. Computer programs in which these methods were implemented (AQ17-DCI and AQ17-HCI) have been tested on several learning problems (Arciszewski et al., 1992; Bala et al., 1992; Wnek, 1993) and have produced promising results. However, the success of a particular method depends on the user's expertise in matching the method's capabilities to the requirements of the learning problem. Therefore, the problem arises of how to automatically adapt a constructive induction method to the problem at hand.

3.3 THE AQ-HCI STRATEGY FOR HYPOTHESIS-DRIVEN CONSTRUCTIVE INDUCTION

The AQ-HCI strategy for Hypothesis-Driven Constructive Induction is described here with some detail because it has been recently investigated by the authors and appears very promising for engineering design purposes (Arciszewski et al. 1994). Therefore, there is a possibility that this strategy will be the first one to be used for actual design purposes.

A hypothesis-driven strategy proposes changes the representation space by analyzing the hypotheses generated in the previous step of the learning process. The initial hypotheses can be generated by a decision rule learning algorithm or some other method. In our experiments, the best results in terms of the predictive accuracy and simplicity of the knowledge acquired have been consistently obtained when using an AQ-type rule learning system (Michalski et al, 1986; Wnek et al, 1995). Some of these results have been described in (Wnek and Michalski, 1994a). Below is a brief description of the AQ-HCI method that uses an AQ-type learning program.

The method works iteratively (Fig. 2), and is divided into two major phases. The first phase determines the best representation space by performing an multistage iterative process of representation space transformation and formulation of intermediate, tentative hypotheses. The second phase determines the final concept description.

In the first phase, each iteration takes training examples (P) projected to the current representation space, identifies descriptions of the concepts to be learned, and then analyses the descriptions to determine what changes, if any, are to be made to the representation space. The process stops when the generated hypothesis is satisfactory or the allocated time or space resources have been exhausted (the “stopping criterion”). Transformations of the representation space may involve both *contraction* and *expansion* operators. Contraction operators decrease the number of attributes spanning the representation space, or the number of values in the attribute value set (by merging values into more abstract units). Expansion operators generate new attributes or add new values to the legal value sets of the existing attributes.

New attributes are generated by detecting patterns in the concept description generated in a given step. By a pattern we mean a component of a concept description that covers a significant number of positive training examples and only a small number of negative examples. The AQ-HCI method searches for the following four types of patterns: value-patterns, condition-patterns, rule-patterns, and class-patterns. Value-patterns aggregate subsets of co-occurring attribute values into single, more abstract values. Condition-patterns represent a conjunction of two or more elementary conditions that frequently occur in a hypothesis. A rule-pattern consists of a set of rules. Class patterns represent relations that a common for subsets of learned classes (concepts) (Wnek, 1993; Wnek & Michalski, 1994ab).

The AQ-HCI strategy was used both as one of the basic strategies as well as to control strategy selection in the multistrategy constructive induction system AQ17-MCI (Bloedorn, Michalski and Wnek, 1993).

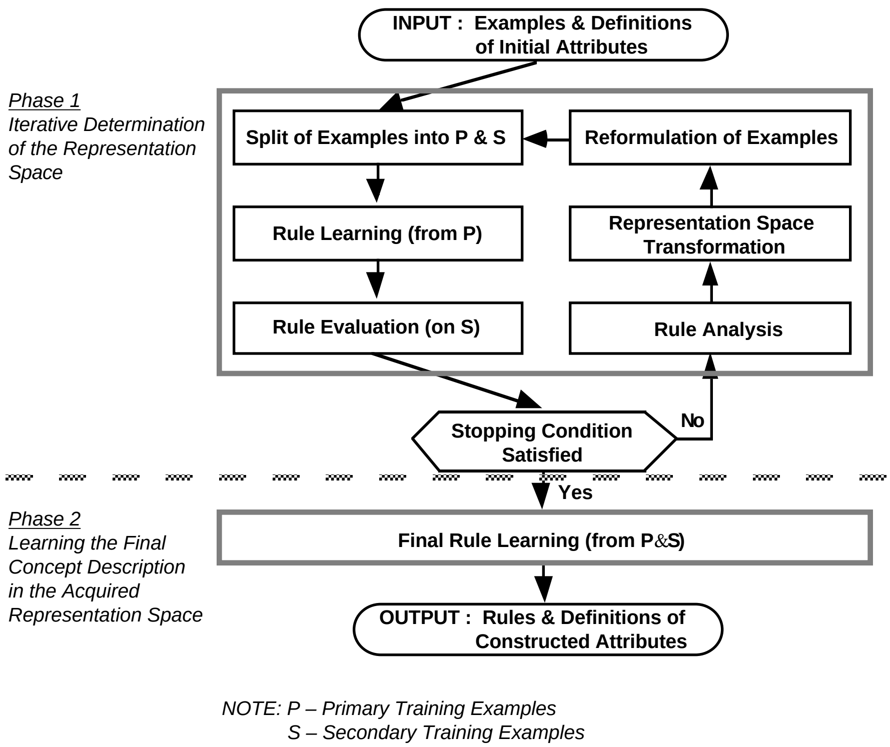


Figure 2. Hypothesis-driven constructive induction: Algorithm.

4. Design Creativity in the Context of Constructive Induction

4.1 INDUCTIVE LEARNING VERSUS CREATIVE DESIGN

There is an important similarity between learning and creative design. Learning can be viewed as a process whose objective is to acquire new knowledge. Such a process includes the acquisition of new facts, the acquisition or discovery of new concepts, the combination of known concepts in order to develop more complex concepts, the determination of relationships among known and newly introduced concepts, etc. Creative design can be viewed as a process whose objective is to produce new concepts in engineering designs. This includes discovering new concepts, combining known concepts into more complex concepts, and, as a byproduct, finding out new relationships among known and newly introduced concepts (knowledge acquisition aspect). Therefore, creative design can be considered as a specific case of a learning process conducted in an engineering domain whose objective is to produce a specific class of engineering concepts. Such understanding enables one to consider creative design in the context of machine learning, particularly in the context of constructive induction, which appears to be the form of learning most relevant to creative design. The reason for this is that constructive induction is concerned with transformations of the representation space.

Constructive induction is the key to understanding creative design because of five major reasons: it 1) generates new attributes, 2) stimulates human thinking, 3) changes the design graphical representation, 4) improves the performance of the designer, 5) allows the formal measuring of the creativity level. Individual reasons are discussed below.

In constructive induction, new attributes are generated that can become *emergent concepts.* Such concepts can lead to a new understanding of a given domain and can produce additional knowledge, including new concepts and their relationships to the primary concepts. Therefore, such new knowledge is relatively easy to comprehend and accept by designers.

The idea of an emergent concept is explained using two examples of the application of the hypothesis-driven constructive induction (AQ-HCI) to a concept learning problem. The first problem is derived from computer science, and is considered mostly for illustrative purposes. It deals with detecting symmetries in concepts and formulating “counting attributes” (e.g., M-of-N concepts). As symmetries often occur in engineering design, the example may be of interest to this area. The second problem concerns a structural design of wind bracings in steel skeleton structures of tall buildings.

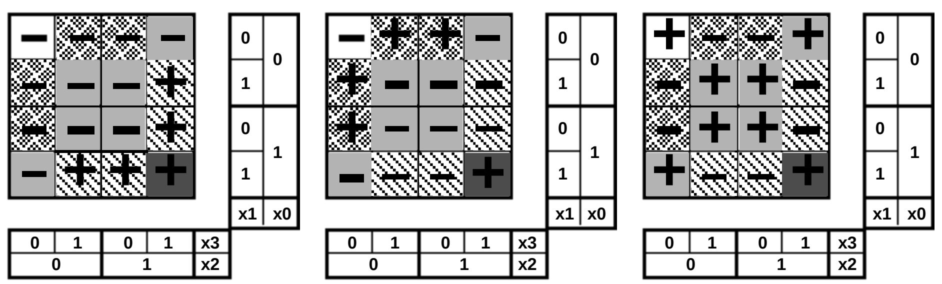
4.2 PROBLEM 1: LEARNING M-OF-N CONCEPTS

This problem concerns a class of learning tasks for which conventional symbolic methods typically produce DNF-type (disjunctive normal form) decision rules that are very long and inaccurate. The essence of the problem is that these methods cannot simply represent concepts that involve counting the presence of some properties in an object. An example of such a "counting property" is the M-of-N concept (“at least M out of N properties of a certain kind are present in an object”). Problems of this type occur in various real-world situations, for example, in medicine (Spackman, 1988), planning (Callan and Utgoff, 1991), game playing (Fawcett and Utgoff, 1991), biology (Baffes and Mooney, 1993) and biochemistry (Towell and Shavlik, 1994).

Fig. 3 gives three examples of M-of-N concepts for an abstract domain: "at least 3-of-4," "1 or 4-of-4," and "even-of-4" (Wnek and Michalski, 1994b). These concepts are symmetrical with regard to input variables (variables can be exchanged without changing the concept). In Fig. 3, these concepts are represented using the diagrammatic visualization tool DIAV (Wnek, 1993, 1995) based on General Logic Diagrams (GLD) proposed by Michalski (1973, 1978b). Each diagram represents a 4-dimensional representation space defined by four binary attributes: x0-x3. Pluses and minuses represent positive and negative instances of concepts, respectively.

In this case, the AQ-XOR (exclusive or) method was used. In the method, arithmetic-type concepts are produced, or they emerge, from logic-type concepts through the detection of *exclusive-or (XOR) symmetry patterns* among pairs of attributes and a subsequent application of the *counting attribute generation rule* of constructive induction. The underlying idea is related to the representation of arithmetic operations using binary logic circuits in computer engineering.

In the method, a combination of two logical operators, AND and XOR, yields a more powerful arithmetic operator ADD. Consequently, new attribute can be defined as an arithmetic sum of attributes. Values of the counting attribute represent the number of attributes that hold for a given concept example, and its domain (value set) is the set of integer values from 0 to N. For all M-of-N concepts in the original representation space, a simple transformation leads to one counting attribute #AttrIn{x1,x2,...xN}. Such an attribute is read "the number of attributes in the attribute set {x1,x2,...xN}."

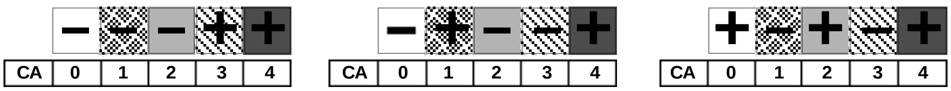


A. At least 3-of-4 **B.** 1 or 4-of-4 **C.** Even-of-4

The shading reflects symmetrical instances in each concept representation.

Figure 3. Three examples of M-of-4 concepts.

Figure 4 shows a visual representation of the three concepts using the newly constructed attribute, CA=#AttrIn{x0,x1,x2,x3}. CA has five values, 0 to 4, that can express the number of properties of the original four attributes {x0,x1,x2,x3}.



A. At least 3-of-4 [#AttrIn{x0,x1,x2,x3} >= 3]

B. 1 or 4-of-4 [#AttrIn{x0,x1,x2,x3} = 1 or 4]

C. Even-of 4 [#AttrIn{x0,x1,x2,x3}=0 or 2 or 4]

By employing the derived concept "number of attributes in the set {x0,x1,x2,x3}" the representation of M-of-N concepts from Figure 3 is straightforward and simple.

Figure 4. Concepts from Figure 3 in the transformed representation space.

The shading of cells matches the shading of respective areas in the original representation space. Up to six cells in the original representation space are mapped into one cell in the space defined by CA. In sum, the emergent concepts that count the number of attributes in the sets of attributes are general. If such concepts are used as new attributes for expanding representation space, then it is simple to represent any M-of-N—type concept.

4.3 PROBLEM 2: STRUCTURAL DESIGN OF WIND BRACINGS

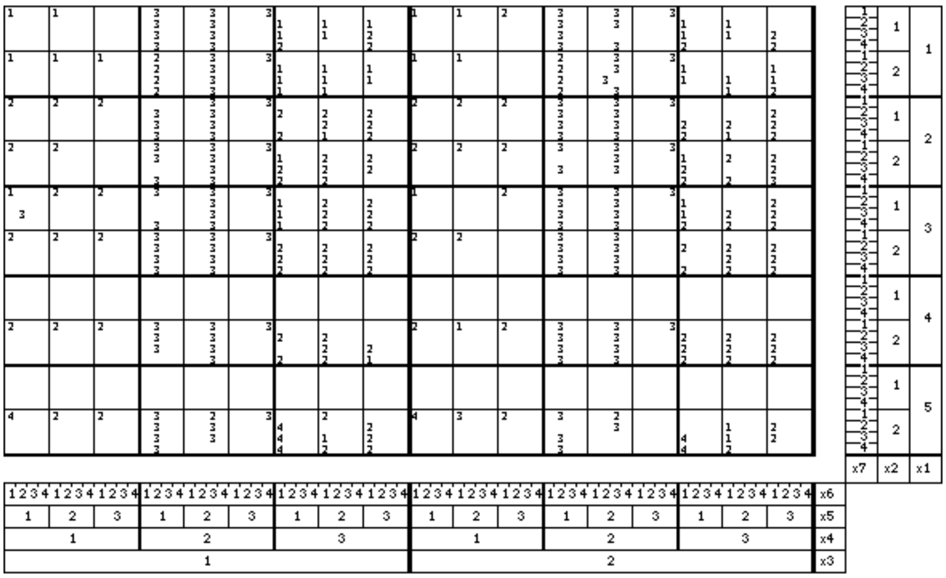
The second example of the application of the AQ-HCI strategy concerns a problem from the area of conceptual design of wind bracings in steel skeleton structures of tall buildings. The objective of the feasibility study reported here was to find decision rules which would assist a designer during the conceptual design stage. These rules represent the relationships among attributes describing the design requirements to be met, the possible structural design decisions, and an assumed quality criterion (in our case - the unit steel weight, as discussed below). The quality criterion was considered a dependent attribute, while all remaining attributes were assumed independent. Decision rules sought were expected to be design rules, or design knowledge, which would show how various structural design decisions taken under different combinations of design requirements would result in one of four values of the quality criterion. Therefore, four categories of design rules were sought, associated with the individual values of the dependent attribute, in our case of the quality criterion. The original representation space consisted of seven independent multivalued attributes: number of stories (x1), bay length (x2), wind intensity factor (x3), type of joints (x4), number of braced bays (x5), number of vertical trusses (x6), and number of horizontal trusses (x7). The first three attributes were used to define design requirements, while the remaining attributes are used to characterize structural design decisions available. Classification of design examples into the four categories of the dependent attribute, called unit steel weight (x8), was done according to the relative unit steel weight, and the following category names were used: low, medium, high, and infeasible for a given design case when a structural system of the assumed type could not be produced. The relative unit steel weight was determined considering all normalized unit weights of various types of wind bracings of the same height designed under identical conditions. Accordingly, design rules related to the dependent attribute category “low” were called “recommendation rules,” those related to the category “medium” were called “standard rules,” and those related to the category “high” were called “avoidance rules.” All rules related to the category “infeasible” were called “infeasibility rules” since they represent relationships among independent attributes which occur in the case when it is impossible to design a wind bracing of a given type under assumed design conditions. The study was conducted using the set of 384 optimal (minimum weight) designs of wind bracing in steel skeleton structures of tall buildings. More details of the design problem considered and the representation space used are provided in (Arciszewski et al. 1994).

Fig. 5 illustrates the knowledge representation space using diagrammatic visualization. Points in this space marked 1, 2, 3, 4 represent individual categories of examples. The used representation space allows for 2880 different possible examples in this space.

For the purpose of the domain considered, the AQ-3S (SubSpace Search) method was used. The AQ-3S method does not extract patterns from hypotheses. Instead, based on an analysis of the hypotheses, it determines which attributes may create strong conjunctions (SCs) and generates concept descriptions from examples projected into subspaces of the original representation space. Subsequently, the rules of such generated descriptions become parts of the new attribute's description. By generating descriptions in subspaces of the representation space, the impact of both attribute noise and classification noise are significantly reduced. This way, strong conjunctions are not split in the process of rule induction.

In the process of expanding the representation space, one multivalued attribute is constructed. For each concept learned, one attribute value is assigned and defined by a set of strong conjunctions characterizing the concept in subspaces of the original representation space. Such definitions combine rules that firmly discriminate the respective class from all other classes. An additional attribute value is assigned to be the negation of all SCs used in defining learned concepts (Szczepanik, Arciszewski and Wnek, 1995).

Fig. 6 shows one of the subspaces considered for detecting discriminant conjunctions. Empty cells represent conditions that were not encountered in the training data. Cells with a single number inside represent conditions that discriminate the given class from all other classes. Cells with multiple numbers represent non-discriminant conditions. Examples described by such conditions have to be characterized using different combinations of original attributes.



Each cell in the diagram contains 16 invisible cells, each representing one vector—a combination of values of seven attributes x1, x2, ... x7 that span the representation space. Numbers 1,2,3,4 marking individual cells denote concepts assigned to the corresponding vectors (also called events). The total representation space consists of 2880 cells. Empty cells represent events for which no concept has been assigned.

Figure 5. Diagrammatic visualization of four concepts (1-4) in structural design.

The constructive induction process produced new useful concepts. The strongest conjunction in the case was [x1=2,3,4] & [x4=2], which accounted for 25% of training examples and uniquely characterized examples of class No. 3, i.e. a class which contains the avoidance design rules. The concept described by [x1=2 or 3 or 4] & [x4=2] is well-known in structural engineering. It is the concept of *truss bracings in mid-height skeleton structures* (12 - 24 stories). The decision rule says that in the USA designing of truss wind bracings for mid-height buildings will result in the relatively high unit weight of bracing and therefore it should be avoided. However, such a rule most likely would never be derived from examples prepared in Europe, where requirements regarding the lateral stiffness of wind bracings are more stringent than in the USA, and therefore truss wind bracings in mid-height buildings are usually preferable, in terms of the unit steel weight, to bracings in the form of rigid frames, or braced rigid frames.

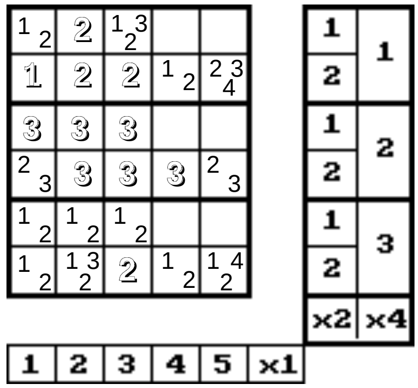


Figure 6. The distribution of examples of four concepts in the representation space defined by x1, x2, and x4.

4.4 CONSTRUCTIVE INDUCTION IN DESIGN

The major objective of constructive induction is to improve the performance of a learning system in terms of its ability to classify objects. When the cooperative design is considered, any improvement in the performance of a design tool immediately results in the improvement of the performance of the entire human-tool system and may be amplified by the human ability to learn. Also, the performance of a learning tool can be formally monitored using various empirical error rates and, therefore, the progress of the entire design “team” can be quantified. This aspect is important in the case of industrial applications of constructive induction.

Constructive induction can also be considered in the context of knowledge acquisition, which is important in design, particularly when knowledge-based support tools are used and continually updated to reflect the changes in the understanding of a given domain. Traditionally, knowledge acquisition in engineering is understood as a process of acquiring design rules. However, equally important is learning a system of concepts in a given design domain; only constructive induction has the unique ability to produce new concepts in the form of constructed attributes.

Learning based on constructive induction may be conducted in parallel with diagrammatic visualization of the representation spaces and all design examples considered, as illustrated in Figs. 5 and 6. In this way, the designer gradually builds his/her understanding of the learning process in terms of changes in the representation space and clustering of examples. The significance of this insight cannot be underestimated from the cognitive point of view. Also, monitoring the subsequent diagrams can be used to determine the progress of learning and when this process is completed. The complexity of emergent concepts/constructed attributes can then be controlled in constructive induction by the "stopping criterion." This criterion requires that the prediction accuracy of the learned concept descriptions exceeds a predefined threshold or that there is no improvement of the accuracy over the previous iteration.

It has already been demonstrated (McLaughlin, 1993) that the discovery of e*mergent values,* which is referred here as e*mergent concepts,* is crucial in creative design for a combination of reasons. In the case of cooperative design, when a human designer is working with a design support tool, an emergent concept is a strong stimulant which can trigger human creativity and may directly lead to creative and patentable designs, as described below.

In the mid-eighties, a series of experiments with a computer program for the generation of design concepts was conducted (Arciszewski, 1988a). The objective of the experiments was to produce an innovative concept or concepts. If this would not be possible, the secondary objective was to obtain some initial results which would stimulate the development of innovative concepts. The program randomly generated combinations of attributes and their values from a given representation space, and was based on the principles of morphological analysis. The first author, who conducted the experiments, had some experience related to the subject of research, including the actual design experience, and was able to determine the structural meaning of results.

The experiments were performed in the area of design of joints in steel space structures, which are mostly used for large span roof structures in exhibition halls and industrial buildings. A joint in a space structure is a geometrically invariable system of connected members (Arciszewski and Uduma, 1988) whose functions are 1) to connect at least three space structure members at a point, 2) to provide the required distribution of external forces applied at this point, and 3) to provide the distribution of the internal forces in individual members connected at this point. In the conceptual design of joints, three major interrelated features are usually considered: the size of a joint, its strength, and its weight. There are strong antinomies among these features, and any improvement in one feature causes undesirable changes in the other two.

In the experiments a 40-attribute representation space was used which was developed over a period of several years in close cooperation with a group of structural designers specializing in steel space structures. The representation space was initially prepared in the late seventies for the patent studies and for practical design purposes. As a result of the experiments, a large class of combinations of attributes and their values was generated and evaluated. The majority of the produced combinations appeared meaningless and was eliminated. However, several combinations were found meaningful from the structural viewpoint, because they actually represented concepts of joints. These concepts were then analyzed in terms of their feasibility and innovation. None of them was evaluated as a success in the search for innovative joints. However, one concept caught attention of one of the authors (Arciszewski) and was found interesting. It inspired him to consider the class of known spherical joints in the context of the generated concept in order to develop a new joint concept which would be feasible and sufficiently innovative to justify its patenting. Thus, it became an emergent concept. From the structural point of view, this concept represented a system of four spheres of equal diameters connected together so that their centers are situated at the vertex of an equilateral pyramid. Such a system is geometrically invariable, light and very rigid, but it is not “smooth,” and it would be difficult to connect structural members to it. The critical issue was how to improve “the smoothness” of the joint while attaining its other desirable structural characteristics. In this case the emergent concept stimulated thinking about spherical joints and about spheres in general, and that led to another interesting and potentially useful concept of a joint. An additional sphere of larger diameter was added to the generated system of four spheres so that a system of five spheres was created. In this case, the spherical joint is in the form of five spheres with four spheres of identical diameter situated inside a fifth one having a larger diameter (Fig. 7). The created joint has a smooth surface (single sphere), can be built with a larger diameter, and may be relatively rigid and light due to the internal bracing in the form of four spheres. Such a joint could be useful for huge space structures, when a great number of heavy members of large diameters requires joints of significant dimensions, which, at the same time, should be smooth outside, simple in form, light, and have advantageous damping characteristics. The concept of this new joint was patented in the USA and Canada.

The experiments with the generation of design concepts by a computer led to the conclusion that the control of the complexity of the *emergent concepts* is important and may lead to planning creative designs with the assumed level of creativity. That control is critical in the case of applied innovative design in industry, when the design objective is sometimes to produce concepts only marginally more innovative than existing patented concepts in order to avoid use of patents and to utilize the available experience. The related issue is the acceptability for designers of the level of complexity of constructed attributes/emergent concepts which can be partially controlled in the constructive induction process, as discussed below.

The above is particularly important in the context of knowledge acquisition in design, especially when knowledge-based support tools are used and continually updated to reflect the changes in the understanding of a given domain. For all these reasons, the control of complexity of the emergent concepts is a new and important design research issue that should be addressed in order to develop design tools with built-in complexity control mechanism. Gregory (1986) divided conceptual design products (design concepts or simply designs) into routine and creative designs, assuming that the former are known while the latter are unknown, yet feasible. Maher and Gero (1993) divided designs into routine, innovative, and creative designs, considering known designs as routine, innovative designs as those with values of the design variables outside the commonly used range, and creative designs as those resulting from the use of new design variables. Altschuller (1969) and Arciszewski (1988b) proposed to divide the designs into five categories, including standard, modification, innovation, invention and discovery.

Figure 7. Patented invention inspired by a computer-generated concept.

Altschuller's classification is based on the nature of knowledge used to produce designs. These examples illustrate the interest of many design researchers in a formal classification of designs and its importance for design research and practice. However, any inflexible classification of designs is inadequate in the context of the Inferential Design Theory. This is particularly true considering that in an actual industrial environment designs are always evaluated in relation to the other known designs. In addition, the designers usually want to determine in a quantifiable way the relative creativity level of a given design. The affiliated problem is how to determine conceptual design operators which have been, or could have been, used to transform the initial known reference design into a different design being considered. In other words, the problem is how to learn what innovative design shaping concepts were used in order to store them for the future utilization. Fortunately, constructive induction provides a conceptual and formal outline for dealing with both problems.

In structural engineering, the morphological distance was proposed in (Arciszewski and Kisielnicka, 1977, Arciszewski, 1986) for measuring the relative complexity, and indirectly the relative creativity, of a structural concept with respect to a certain reference concept. This distance was defined as the number of different values of attributes describing both the concept under consideration and the reference concept. Similar ideas have been used in constructive induction for controlling the extent of building constructed attributes, which is based on the concept of the logical distance, proposed by Michalski (1975).

The distance between two events is defined as the sum of the distances between values of descriptors used in describing the events. A measure of the distance d(x, y) between the values of a descriptor depends on the type of descriptor. Three types of descriptors are considered: interval, nominal, and structured. The distance d(x,y) between two descriptor values, x and y, is defined differently for different descriptor types:

1. Interval descriptors

$$d(x,y) = \frac{|x-y|}{card(X)}$$

where x and y belong to the interval represented by the set X: {0, 1, 2, ..., n}. card(X) is the cardinality of the set, i.e. the size of the interval.

2. Nominal descriptors

$$d(x,y) = \begin{cases} 1, & \text{if x is not identical to y} \\ 0, & \text{otherwise} \end{cases}$$

3. Structural descriptors

$$d(x,y) = \frac{NB}{MNB}$$

where NB is the length of the shortest path linking x with y and MNB is the length of the longest of all the shortest paths linking any two nodes in the structure.

The measure of the logical distance can then be used to determine the relative creativity level of a given concept for the purpose of its evaluation and it can also be used for the control of the constructive induction process in order to achieve an assumed level of creativity. To illustrate this idea of measuring the relative creativity of a given concept through the use of the logical distance, an example from the area of conceptual design of wind bracing in steel skeleton structures is provided. For the clarity of the example, only four simple transverse wind bracings in a three-bay symmetrical skeleton structure are discussed. A more detailed analysis of the conceptual design of wind bracings in tall buildings in the context of the morphological distance can be found in (Arciszewski 1986).

In the design case presented, the simplest wind bracing is in the form of a single one-bay rigid frame centrally located in the skeleton structure. Therefore, it can be used as a reference concept. This concept is described by the attributes x4 = 1, x5 = 1, x6 = 1, x7 = 1 (for the discussion of these attributes see section 4.3, Problem 2: Structural Design of Wind Bracings) and it is denoted by No. 1 in Fig. 8.

A large class of innovative wind bracings with respect to this one can be developed and the relative creativity of individual concept can be formally evaluated using the logical distance. The reference concept of a single one-bay frame can be easily developed into two similar concepts of two one-bay rigid frames and of a three-bay rigid frame. The first one is described by the attributes x4 = 1, x5 = 2, x6 = 1, x7 = 1 and denoted by No. 2 in Fig. 8 while the second one is described by the attributes x4 = 1, x5 = 3, x6 = 1, x7 = 1 and denoted by No. 3 in the same figure. The new concepts No. 2 and 3 differ from the reference concept No. 1 in one attribute only (attribute x5 - the number of bays entirely occupied by the wind bracing). Therefore their relative creativity with respect to the concept 1, as measured by the logical distance, is equal to unity and it is obviously low. This is a typical case of an innovation in accordance to Maher and Gero (1993). However, the initial reference wind bracing can be also developed adding a horizontal truss situated on its top, as shown in Fig. 8 for the concept denoted by No. 4. This concept is described by the attributes x4 = 3, x5 = 3, x6 = 1, x7 = 2. In this case, values of two attributes are different than for the reference concept, therefore the relative creativity of this concept in terms of the logical distance is two, and its is obviously higher than in the case of concepts No. 2 and 3. From the structural point of view, this result is also consistent with the intuitive understanding of the relative complexity if this new concept, which was produced making two design decisions regarding the nature of joints and the number of horizontal trusses in bracing, as opposed to the single design decision related to the number of bays occupied by bracings which let to the Concepts No. 2 and 3.

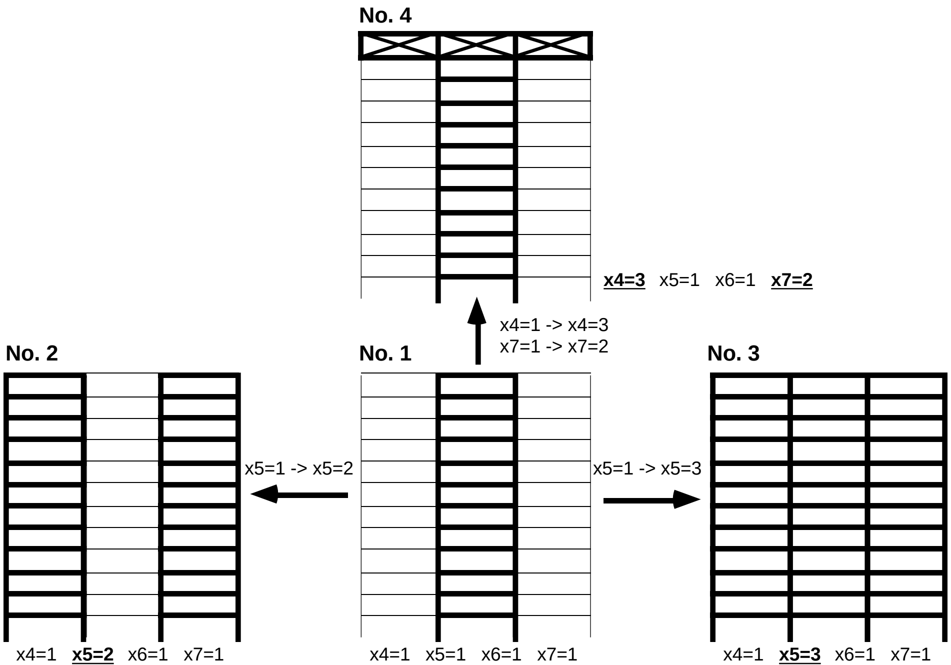


Figure 8. Logical distance of wind bracing concepts.

The development of a creativity measure for structural design concepts was the subject of the research on the design of joints in steel space structures (Arciszewski and Uduma, 1988). It resulted in the identification of six "innovative shaping concepts," or conceptual design operators, which were used to produce a class of 15 patented spherical joints in steel space structures. All these joints were related to the first spherical joint in space structures, called MERO, which was patented in 1935. The identified innovative shaping concepts included the elimination of material, or part of the joint, the use of division or multidivision of the joint, the use of symmetry, the use of asymmetry, the addition of internal or external components, and the shape change of the joint or of its end-pieces. In this case, the relative creativity of individual joints was proposed to be measured by the "innovative distance" from the assumed reference MERO joint considering the number of innovative shaping concepts necessary to transform the MERO joint into a given joint. All identified innovative shaping concepts, as well as the innovative distance, could be easily expressed in terms of attributes describing the joint. Therefore, they could be considered as operators whose use resulted in new concepts (constructed attributes) which are based on primary concepts. In this way, constructive induction can be considered here as an underlining computational foundation for conceptual design whose objective is to produce creative concepts.

5. Conclusions

Constructive induction has been demonstrated to be a powerful theoretical framework for describing an engineering design process, in particular, a creative conceptual design. Several crucial ideas, such as an emergent design concept and a task-oriented improvement of the design representation space, have been presented in terms of constructive induction. Their interpretation and use is consistent with both the areas of design and computer science.

Constructive induction and its system of concepts and methods have been found highly relevant to explaining conceptual design. Therefore, constructive induction has been proposed as a new paradigm for developing computational foundations for conceptual design. The control of the creativity level of design concepts can be accomplished using various proposed complexity (or relative creativity) measures. The generation of emergent concepts in constructive induction, and their explicit presentation to the designer, was found to be particularly important because of cognitive, computational, and engineering reasons. In general, the introduction of constructive induction to design bridges the gap between engineering design and artificial intelligence. This result may naturally lead to the development of a new generation of design support tools with a real impact on design practice.

Further research in the presented direction includes the development of theoretical foundations and practical methods for the constructive induction-based conceptual design. Various design experiments using constructive induction methods will be initiated. In view of the existence of several constructive induction strategies and methods, future research will explore the usefulness of multistrategy constructive induction methods in conceptual design. Such methods integrate various combinations of strategies. A multistrategy constructive induction system for design will be experimentally developed and ultimately turned into a practical design support tool.

Acknowledgments

Authors thank Eric Bloedorn, John Doulamis and Ken Kaufman for discussions and help in the development of some of the ideas presented in this paper.

This research was conducted in the Machine Learning and Inference Laboratory at George Mason University. The Laboratory’s research is supported in part by the National Science Foundation under Grants No. CDA-9309725, IRI-9020266, and DMI-9496192, in part by the Office of Naval Research under Grant No. N00014-91-J-1351, and in part by the Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, the Grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research.

References

Altschuller, H.: 1969, *Algorithm of Invention*, Moskowskij Raboczij Publishing House, Moscow.

Arciszewski, T. and Kisielnicka, J.: 1977, Morphological Analysis, *Problem, Method, Solution, Methods of Creative Problem Solving Vol. I*, A. Goralski (ed), WPN Publishing House, Warsaw.

Arciszewski, T.: 1986, Decision Making Parameters and Their Computer-Aided Analysis for Wind Bracings, *Advances in Tall Buildings*, L.S. Beedle (ed), Van Nostrand Reinhold Company, New York, pp. 281-298.

Arciszewski, T.: 1988a, Stochastic Form Optimization, *Journal of Engineering Optimization*, **13**, 17-33.

Arciszewski, T.: 1988b, ARIZ 77: An Innovative Design Method, *Journal of Design Methods and Theories*, **22**(2), 796-821.

Arciszewski, T., Bloedorn, E., Michalski, R.S., Mustafa, M. and Wnek, J.: 1994, Machine Learning of Design Rules, *Journal of Computing in Civil Engineering*, **8**(3), 286-309.

Arciszewski, T. and Uduma, K.: 1988, Shaping of Spherical Joints in Space Structures, *International Journal of Space Structures*, **3**, 171-183.

Baffes, P.T. and Mooney, R.J.: 1993, Symbolic Revision of Theories with M-of-N Rules. In *Proceedings of the 2nd International Workshop on Multistrategy Learning* Harpers Ferry, WV, Morgan Kaufmann, pp. 69-75.

Bala, J.W., 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*, Aberdeen, Scotland, Morgan Kaufmann, pp. 20-29.

Bentrup, J.A., Mehler, G.J. and Riedesel, J.D.: 1987, *INDUCE 4: A Program for Incrementally Learning Structural Descriptions from Examples*, Reports of the Intelligent Systems Group, ISG 87-2, Urbana-Champaign, University of Illinois, Department of Computer Science.

Bloedorn, E. and Michalski, R.S.: 1991, Data Driven Constructive Induction in AQ17-PRE: A Method and Experiments. *Proceedings of the Third International Conference on Tools for AI.* San Jose, CA.

Bloedorn, E., Michalski, R.S. and Wnek, J.: 1993, Multistrategy Constructive Learning: AQ17-MCI. *Proceedings of the Second International Workshop on Multistrategy Learning MSL-93*, Harpers Ferry, WV.

Callan, J.P. and Utgoff, P.E.: 1991, A Transformational Approach to Constructive Induction, *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, Morgan Kaufmann, pp. 122-126.

De Raedt, L. and Bruynooghe, M.: 1989, Constructive Induction By Analogy: A Method to Learn How to Learn? *Proceedings of EWSL-89*, Montpellier, France, Pitman, pp. 189-200.

Drastal, G., Czako, G. and Raatz, S.: 1989, Induction in an Abstraction Space: A Form of Constructive Induction. *Proceedings of the IJCAI-89*, Detroit, MI, Morgan Kaufmann, pp. 708-712.

Emde, W., Habel, C.U. and Rollinger, C.-R.: 1983, The Discovery of the Equator or Concept Driven Learning. *Proceedings of IJCAI-83*, Karlsruhe, Germany, Morgan Kaufmann, pp. 455-458.

Falkenhainer, B.C. and Michalski, R.S.: 1990, Integrating Quantitative and Qualitative Discovery in the ABACUS System, *in* Y. Kodratoff and R.S. Michalski (eds), *Machine Learning: An Artificial Intelligence Approach*, Vol. III, Morgan Kaufmann, Palo Alto, CA.

Fawcett, T.E. and Utgoff, P.E.: 1991, A Hybrid Method for Feature Generation. *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, Morgan Kaufmann, pp. 137-141.

Flann, N.S. and Dietterich, T.G.: 1986, Selecting Appropriate Representations for Learning from Examples, *Proceedings of AAAI-86* Philadelphia, PA, Morgan Kaufmann, pp. 460-466.

Flann, N.S.: 1990, Improving Problem Solving Performance by Example Guided Reformulation of Knowledge, *in* D.P. Benjamin (ed), *Change of Representation and Inductive Bias*, Kluwer Academic, Boston, MA.

Gero, J.S. and Maher, M.L.: 1993, Introduction, *in* J.S. Gero and M.L. Maher (eds), *Modeling Creativity and Knowledge-Based Creative Design*, Laurence Erlbaum Associates, Publishers, Hillsdale, New York.

Greene, G.H.: 1988, *The Abacus 2 System for Quantitative Discovery: Using Dependencies to Discover Non-Linear Terms*, Reports of Machine Learning and Inference Laboratory, MLI 88-4, Center for Artificial Intelligence, George Mason University, Fairfax, VA.

Gregory, S.: 1986, *Private Communication.*

Knoblock, C.A.: 1990, A Theory of Abstraction for Hierarchical Planning, *in* D.P. Benjamin (ed), *Change of Representation and Inductive Bias*, Kluwer Academic Boston, MA.

Knoblock, C.A., Minton, S. and Etzioni, O.: 1991, Integrating Abstraction and Explanation-Based Learning in PRODIGY, *Proceedings of AAAI-91*, AAAI Press/The MIT Press, pp. 541-546.

Kokar, M.M.: 1985, Discovering Functional Formulas Through Changing Representation Base, *Proceedings of AAAI-86*, Philadelphia, PA, pp. 455-459.

Langley, P., Bradshaw, G.L. and Simon, H.A.: 1983, Rediscovering Chemistry With the BACON System, *in* R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA.

Langley, P.W., Simon, H.A., Bradshaw, G.L. and Zytkow, J.M.: 1987, *Scientific Discovery: Computational Explorations of the Creative Processes*, The MIT Press Cambridge, MA.

Larson, J.B. and Michalski, R.S.: 1977, Inductive Inference of VL Decision Rules, *invited paper for the Workshop on Pattern-Directed Inference Systems, ACM SIGART Newsletter*, **63**, 38-44.

Lenat, D.B.: 1977, On Automated Scientific Theory Formation: A Case Study Using the AM Program, *in* J.E. Hayes, D. Michie and L.I. Mikulich (eds), *Machine Intelligence Vol. 9*, Halsted Press, New York.

Lenat, D.B.: 1983, Learning from Observation and Discovery, *in* R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA.

Matheus, C.: 1989, *Feature Construction: An Analytic Framework and Application to Decision Trees*, Ph.D. Dissertation, Urbana-Champaign: University of Illinois.

McLaughlin, S.: 1993, Emergent Value in Creative Products: Some Implications for Creative Processes, *Modeling Creativity and Knowledge-Based Creative Design*, *in* J.S. Gero and M.L. Maher (eds), Laurence Erlbaum Associates, Publishers, Hillsdale, New York.

Michael, J.: 1991, *Validation, Verification, and Experimentation with Abacus2*, Reports of the Machine Learning and Inference Laboratory, MLI 91-8, Center for Artificial Intelligence, George Mason University, Fairfax, VA.

Michalski, R.S.: 1973, AQVAL/1 - Computer Implementation of a Variable-Valued Logic System VL1 and Examples of its Application to Pattern Recognition. *Proceedings of the First International Joint Conference on Pattern Recognition*, pp. 3-17.

Michalski, R.S., 1975, On the Selection of Representative Samples from Large Relational Tables for Inductive Inference, *Technical Report No. M.D.C 1.1.9*, Department of Engineering, University of Illinois at Chicago Circle, Chicago.

Michalski, R.S.: 1978a, Pattern Recognition as Knowledge-Guided Computer Induction, *Technical Report No. 927*, Department of Computer Science, University of Illinois, Urbana-Champaign.

Michalski, R.S., 1978b, A Planar Geometrical Model for Representing Multidimensional Discrete Spaces and Multiple-valued Logic Functions, *Technical Report*, Department of Computer Science, University of Illinois, Urbana-Champaign.

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

Michalski, R.S., Mozetic, I., Hong, J. and Lavrac, N.: 1986, The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proceedings of AAAI-86*, Philadelphia, PA, pp. 1041-1045.

Michalski, R.S., "Inferential Theory of Learning: Developing Foundations for Multistrategy Learning," in *Machine Learning: A Multistrategy Approach*, Vol. IV, R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, San Mateo, CA, 1994.

Mitchell, T.M., Utgoff, P.E. and Banerji, R.: 1983, Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics, *in* R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA.

Morik, K.: 1989, Sloppy Modeling, *in* K. Morik (ed), *Knowledge Representation and Organization in Machine Learning*. Springer-Verlag, Berlin.

Muggleton, S.: 1987, Duce, and Oracle-Based Approach to Constructive Induction, *Proceedings of IJCAI-87*, Milan, Italy, Morgan Kaufmann, pp. 287-292.

Muggleton, S. and Buntine, W.: 1988, Machine Invention of First Order Predicates by Inverting Resolution, *Proceedings of the 5th International Conference on Machine Learning*, (pp. 339-352, Ann Arbor, MI: Morgan Kaufmann.

Pagallo, G. and Haussler, D: 1989, Two Algorithms that Learn DNF by Discovering Relevant Features, *Proceedings of the 6th International Machine Learning Workshop*, Ithaca, NY, Morgan Kaufmann, pp. 119-123.

Pagallo, G. and Haussler, D.: 1990, Boolean Feature Discovery in Empirical Learning. *Machine Learning*, **5**, 71-99.

Rendell, L.: 1985, Substantial Constructive Induction Using Layered Information Compression: Tractable Feature Formation in Search. *Proceedings of IJCAI-85*, (pp. 650-658).

Rendell, L. and Seshu, R.: 1990, Learning Hard Concepts Through Constructive Induction: Framework and Rationale. *Computational Intelligence*, **6**, 247-270.

Schlimmer, J.C.: 1987, "Learning and Representation Change," *Proceedings of AAAI-87*, Morgan Kaufmann, pp. 511-515.

Spackman, K.A.: 1988, Learning Categorical Decision Criteria in Biomedical Domains. In *Proceedings. of the 5th International Conference on Machine Learning*, Morgan Kaufmann, San Mateo, CA, pp. 36-46.

Szczepanik, W., Arciszewski, T. and Wnek, J.: 1995, Empirical Performance Comparison of Two Symbolic Learning Systems Based on Selective and Constructive Induction. In *Proceedings of the IJCAI-95 Workshop on Machine Learning in Engineering*, Montreal, Canada.

Towell, G. G. and Shavlik, J. W.: 1994, Refining Symbolic Knowledge Using Neural Networks, *in* R.S. Michalski and G. Tecuci (eds), *Machine Learning: A Multistrategy Approach, Vol. IV,* Morgan Kaufmann, San Mateo, CA, pp. 405-429.

Utgoff, P.E.: 1984, *Shift of Bias for Inductive Concept Learning*. Ph.D. dissertation, Rutgers University.

Utgoff, P.E.: 1986, Shift of Bias for Inductive Concept Learning, *in* R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds), *Machine Learning: An Artificial Intelligence Approach*, Vol. II, Morgan Kaufmann, Los Altos, CA.

Wnek, J. and Michalski, R.S.: 1991, Hypothesis-Driven Constructive Induction in AQ17 - A Method and Experiments, *Proceedings of IJCAI-91 Workshop on Evaluating and Changing Representation in Machine Learning*, Sydney, Australia, pp. 13-22.

Wnek, J.: 1993, *Hypothesis-Driven Constructive Induction*, Ph.D. dissertation. School of Information and Technology, George Mason University.

Wnek, J. and Michalski, R.S.: 1994a, Hypothesis-driven Constructive Induction in AQ17-HCI: A Method and Experiments, *Machine Learning*, **14**, 139-168.

Wnek, J. and Michalski, R.S.: 1994b, Conceptual Transition from Logic to Arithmetic in Concept Learning, *Reports of the Machine Learning and Inference Laboratory*, MLI 94-7, George Mason University, Fairfax, VA.

Wnek, J.: 1995, DIAV 2.0 User Manual: Specification and Guide through the Diagrammatic Visualization System, *Reports of the Machine Learning and Inference Laboratory*, MLI 95-5, George Mason University, Fairfax, VA.

Wnek, J., Kaufman, K., Bloedorn, E. and Michalski, R.S.: 1995, Inductive Learning System AQ15c: The Method and User's Guide, *Reports of Machine Learning and Inference Laboratory*, MLI 95-4, George Mason University, Fairfax, VA, 1995.

Wrobel, S.: 1989, Demand-Driven Concept Formation, *in* K. Morik (ed), *Knowledge Representation and Organization in Machine Learning*. Springer Verlag, Berlin Heidelberg.