

An Expert System for Optimizing Computer Aided Design of Post Frame Buildings

K. G. Gebremedhin, S. S. Jagdale, R. Gupta

ASSOC. MEMBER
ASAE

ABSTRACT

A procedure has been developed to optimize designs of metal-clad post-frame buildings using an expert system. The novel part of the study is the development of a procedure that extends the solution of an engineering problem from a conventional algorithmic Computer Aided Design (CAD) program to be used by the knowledge base of an expert system. The expert system then uses this information to optimize the design based on specified rules. The system shows a procedure for linking and sharing information between CAD and expert system programs.

INTRODUCTION

Recently developed programs for Computer Aided Design (CAD) of structures perform the tedious and complex numerical computations and present the design in graphic forms. In addition, some CAD programs check for the consistency of the design according to design specification and perform error checks. In "interactive" CAD systems, some degree of interaction is provided between the user and the machine (Gebremedhin, 1986; Suddarth and Wolfe, 1984; Adeli and Al-Rijleh, 1987; Adeli and Fiedorek, 1986). In these programs, most, if not all, of the decision making process is left to the user. These computer programs have either no or a very limited knowledge base. In addition, they cannot explain their line of reasoning or justify their answers.

A natural extension of interactive CAD programs would be knowledge-based expert systems (Adeli, 1986). The primary characteristics of expert systems have been delineated in Waterman (1986) and Hayes-Roth et al. (1983). Among them are the separation of the knowledge base (KB) from the inference mechanism, use of heuristics and the explanation facility. Expert systems tend to mimic the decision-making and reasoning process of human experts by providing expert advice, answering questions, and justifying their conclusions. A major advantage is that its modular structure simplifies the development, maintenance, and updating of the expert system (ES). Another advantage of this method is the ease of KB modification. The knowledge base in an ES is maintained as a set of rules which are blocks of knowledge that may be applied independently of one

another (depending on the particular problem at hand). Therefore, addition of more rules or deletion/modification of existing rules can be done very conveniently. This is in contrast with the algorithmic/sequential methodology of conventional programming where any modification involves substantial changes in the source code of the program itself.

In this article, a procedure is presented to optimize designs of metal-clad, post-frame buildings using a knowledge based expert system. This was accomplished by first solving a post-frame structure including diaphragm action using a frame analysis program. Then, the solutions (stresses and deflections) obtained from the frame analysis program were used by the KB of the expert system. The expert system optimizes the design based on specified rules. This procedure extends not only the solution of an engineering CAD problem, but also the concept of communication that is needed to share information between CAD programs (separately available to perform complex calculations and provide results) and expert system programs (that use rules to deduce conclusions on the basis of data provided). The problem has long been the incompatibility of these problems and the lack of communication that is needed to share information between them. The frame analysis program used in this study was METCLAD (Gebremedhin, 1988a).

Objective

The system, henceforth referred to as Met-X-PERT, has been developed on an IBM/PC with a hard disk drive and 512K of RAM. Met-X-PERT is developed in Turbo Pascal. The program is limited to the design of metal-clad, post-frame rectangular buildings that exhibit diaphragm action by the roof envelope. The program is not intended for practical application because of the inherent difficulty of incorporating in program rules all assumptions, requirements, and judgements associated with design. The purpose of this article is, therefore, to develop a procedure for linking and sharing information between conventional algorithmic CAD program and an expert system, so that others developing similar expert systems for other applications would benefit. Besides, it may be used as a teaching tool in timber engineering, computer-aided design, and expert system courses.

DESIGN ALGORITHM

Optimization of individual members of a structure begins after the output of the frame analysis program (METCLAD) is completed. An interface routine extracts the stress and deflection information from the output of

Article was submitted for publication in October 1988; reviewed and approved for publication by the Emerging Technology Div. of ASAE in April 1989.

The authors are: K. G. GEBREMEDHIN, Associate Professor, Agricultural and Biological Engineering Dept.; S. S. JAGDALE, Graduate Student, Mechanical and Aerospace Engineering Dept.; and R. GUPTA, Graduate Student, Agr. and Bio. Eng. Dept., Cornell University, Ithaca, NY.

METCLAD and creates a fact-base. These facts are displayed on the screen.

The stresses and deflections are analyzed based on the rules specified in the knowledge base. The knowledge representation in Met-X-PERT is through production rules. The program supports numeric facts. A numeric fact is a fact with a numeric value. Production rules are the basic building blocks of the knowledge base. The production rules consist of antecedent and conclusion parts. The conditional antecedent part consists of IF, and AND statements. If all the conditional statements are TRUE, then the state of the fact of the THEN statement is added to the working memory. A production rule is expressed as

```

IF
  <condition 1>
AND
  <condition 2>
THEN
  <conclusion>
  
```

The advice and actions taken are displayed on the screen. The actions that may be taken include a change in post size, truss member size, stress grade of lumber, or gauge of the metal skin.

To take any or all of these actions, the program looks at several limiting factors for design, for example, the shear force in the metal skin, the Combined Stress Interaction Index (CSI) value of a truss chord member, deflection, and the CSI value of a post. If there are suggestions or advice to make several modifications in the design (e.g., change metal skin gauge and increase size of a truss chord, etc.), then all these changes are made at once. The only limitation is that the program looks at only one truss chord member and one post at a time. So, if there are, say, three truss chord members which need modification, it takes into consideration only the first one during the present iteration. If this problem

gets solved in one iteration, then during the next iteration, it looks at the next chord. In other words, it looks at one element in a set of similar elements (truss chords are one set of elements, truss web members are another set of elements, posts are yet another set, etc.). The changes triggered by the inference engine modifies the input file to METCLAD. METCLAD is then executed again using the modified input file. A new output file is generated and this output file is similarly evaluated by the expert system. This continues until all limiting factors for design are met and an "optimum" design is simulated. A flow chart which shows the sequence of actions in Met-X-PERT is given in Fig. 1.

Limitations and Assumptions

The following limitations and assumptions are made in developing the rules in Met-X-PERT. These limitations and/or assumptions can easily be changed or edited, if desired.

1. Member size is changed in steps of 2 in. Truss members must have 2-in. "nominal" width. Thus, only the depth dimension changes. Posts should have more than 2-in. nominal width.
2. Stress grade of lumber is changed in steps of 10%. This process continues until the minimum or maximum stress grade of lumber, as specified by the NFPA (1986), is reached.
3. Only two gauges of metal skin (26 and 29) are considered in the program. These are the two gauges commonly used for roof cladding of post-frame buildings. (The fastening type and pattern for the roofing are defined by a prototype diaphragm in the building.)
4. Truss web members are subjected to axial forces only.

ARCHITECTURE OF Met-X-PERT

Met-X-PERT is structured in a modular form for ease of modifications. It consists of 12 modules grouped according to their utilities. The structure and utility of each module is described briefly. Further details of the program PROCEDURES and FUNCTIONS are given in Gebremedhin (1988b).

Declaration

This module contains all the declarations for the entire program. Some of these pertain to the list of members in the structure and their properties, and others to the menu. The declarations are used to interpret user input and to manage objects and values.

Knowledge Base Utilities

This section contains the knowledge base manipulation utilities necessary to develop and maintain a linked object list. An expert system must have a mechanism for inserting facts and rules into the knowledge base, maintaining the list of expressions in the KB, and displaying the contents of the KB for review by the user of the system. The knowledge base in Met-X-PERT consists of 32 rules. The system contains the knowledge necessary to optimize the size of post and truss members, the horizontal deflection at the eave, and shear of the metal skin.

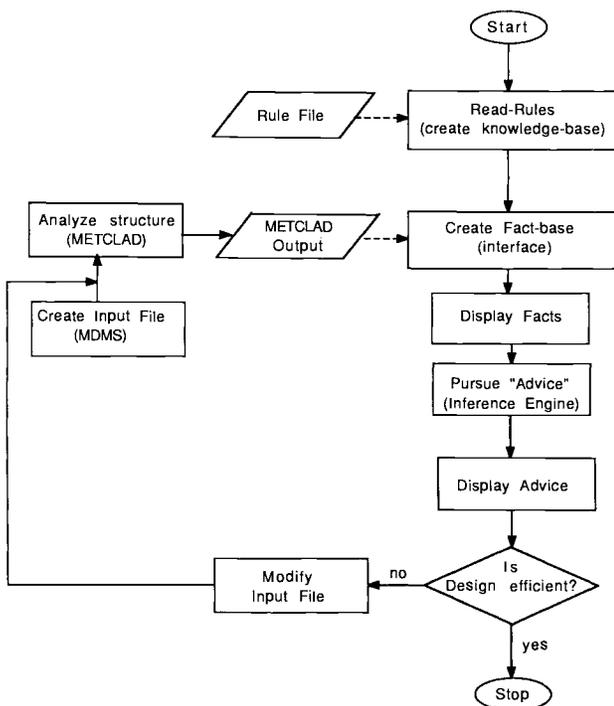


Fig. 1—Sequence of actions in Met-X-PERT.

Module for Handling Variable Certainty

The routines in this module allow the expert system to handle variable certainty. The program reads a certainty factor (cf) integer in an object-value couplet, or alternatively, translates subjective input into quantitative certainty factors on a scale of 0 to 100. The confidence factor or certainty factor is used to reflect the degree of certainty with which a fact is believed. This information is recorded in the fact base along with the object-value couplet describing that fact. If there are several rules which have different condition-element (or antecedents) but similar conclusions, and more than one gets triggered, then the degree of certainty of the conclusion will be greater than any one of them individually. In such cases, the cf's need to be accounted for in a combined way. For this purpose, the concept of BLENDING FUNCTION is used. The blending function is basically an arithmetic expression which combines two certainty factors (cf1 and cf2) and normalizes the result, so that the result lies between 0 to 100. The particular blending function that was used in this program uses a simple smoothing technique to blend the cf's as (Alty and Coombs, 1984; Negoita, 1984):

$$cf_{\text{result}} = \frac{100 * cf1 + 100 * cf2 - cf1 * cf2}{100}$$

where

cf1 and cf2 = certainty factors, percentages.

Additional modules are used to add the cf integer to the linked object list and permit existing cf integers to be updated.

Module for Multivalued Expressions

The routines in this module are used to ascertain whether or not an object in the linked list is multivalued and to define a particular object name as being multivalued so that more than one value can be assigned to the given object name. For example, *advice* is a multivalued object.

Module for Questions and Legal Values

The routines in this section enable the expert system to interpret operator input containing an object name and a list of legal values to store the value names in the linked object list. The program is also equipped with a facility for adding questions about specified objects and displaying existing questions along with the associated legal values. This approach is very helpful for interactive consultation type expert system applications.

Rules Management Module

In this section, there are routines which enable the expert system to read, save, and retrieve rules of the form:

```
Rule
IF
  metal skin = gauge 29
AND
  shear force of metal skin = less than the allowable
AND
  CSI of post = greater than 1.0
THEN
```

recommended action = increase gauge of metal skin to 26

There are also PROCEDURES to perform simple house-keeping chores, add a premise to the current rule, add a conclusion to the current rule, and display an existing rule on request.

Explanation Module

The system guides and assists the user in the design decision-making process by providing explanations. The PROCEDURES in this section allow the expert system to explain its line of reasoning, how a specific conclusion was reached, and why a certain question must be asked to pursue the stated goal. The user can review the rules of the knowledge base and trace the line of reasoning. This procedure not only demonstrates the program's approach to solving a problem but also permits the operator to monitor the system as it processes the user-constructed rules.

Module for the Inference Engine

The inference engine applies its repertoire of rules in an attempt to assign a value to the named object-goal. The program continues its pursuit until one of the candidate solutions succeeds. A backward chaining logic was followed. In this, the engine begins with a terminal conclusion, the object-goal. The program then searches through the rule list for any rule that has the object-goal as its conclusion. The engine then tests each part of the rule's premise and the process is begun again for the premises.

At the heart of the program's inference engine is the PROCEDURE *pursue*. It acts on an object-name, trying to apply the rules to give the object a value. If no rules conclude anything about the object, the user is asked to enter value of that object. The PROCEDURE uses a rule pointer to point to the current rule under consideration. The program next sets out to find a rule that has the object name in its conclusion. It then tests each part of the rule's premise. If all the premises are satisfied, the conclusion is added to the knowledge base. If the object's value cannot be found from the rules, the program prompts the user to enter a value name. Other PROCEDURES exist which implements a conclusion when the inference engine finds a link between the premise of one rule and the conclusion of another. The procedure ascertains the cf of the conclusion based on the relative cf integers of the contributing rules and prints its conclusion on the screen.

Interface Module

This module interfaces the METCLAD output with Met-X-PERT. Information extracted from METCLAD output include: (a) combined stress interaction indices of post and truss members, (b) horizontal shear stresses of post and truss members, (c) shear strength of the metal skin, and (d) horizontal deflection at eave height. This information is stored and the limiting factors for design are analyzed. Routines exist for initializing the variables from the METCLAD output, and converting a character string representing a number to its actual numeric value.

Analyzer Module

This module analyzes the limiting factors for design based on the rules specified.

Output Interface Module

This module creates a new file called "expert.MCI" depending on the advice and the initial values of the problem specifications. This file can be used directly as input to the METCLAD program.

Main Program

This program integrates together the different modules in the system to perform their functions.

OPERATIONAL PROCEDURE

This section describes the software requirements and the procedures required to operate the system. It is not meant to be an operator's manual for the program developed but to show the sequence of actions that need to be taken and to also show the information displayed at each step.

Software Requirements

The program diskette contains the following files:

1. MDMS.COM - preprocessor to create a data file. MDMS is an acronym for METCLAD Data Management System;
2. fn.MCI - data file name created with MDMS. This file could have any name acceptable in DOS but should have MCI for its extension;
3. METCLAD.EXE - analyzes the data file created by MDMS;
4. Met-X-PERT.EXE - optimizes METCLAD output;
5. RULES - contains the rules in the knowledge base.

Procedure

To operate the program, the following steps are followed (the boxed statements are what you see on the computer's monitor):

1. Create a data file (e.g., fn.MCI) using the preprocessor, MDMS. Further details about this system is given in Gebremedhin and Bartlett (1987). When creating a new file, it is advisable to have a back-up file with a different extension. This is because only the file with an MCI extension is executed by the expert system, and this file is constantly modified based on the advice invoked. The back-up file with the different extension can not be executed by the expert system.

2. Type Met-X-PERT <return> and the screen displays

Welcome to the Met-X-PERT

Press ENTER to continue.

3. Press <return>, and the screen displays

--> Enter file name without extension:

4. After the file name is entered, the screen displays

--> Enter connectivity of roof (1 or 2)

5. Connectivity of roof refers to the condition of continuity of the metal skin at the ridge of gable roofs. Enter 1 if the metal skin at the ridge of a sloped roof is assumed continuous for shear transfer, otherwise enter 2, and press <return>. The screen displays

Executing METCLAD....

Iteration no. = 1

fn 1

When execution of the data file is completed, the program creates an output file called fn.MCO. The following sequence of execution are performed and the messages are displayed on the screen:

--> METCLAD execution completed.

--> Reading the file containing rules...

--> Reading the METCLAD output...

When the above actions are completed, the screen displays a list of the potential limiting factors pertaining to the problem being analyzed as

--> Analyzing METCLAD output...

- Gauge of metal skin: 29

- Shear force of skin = 3.1710E+03

- Allowable shear force of skin = 4.3280E+03

- Critical Post is: 2-4

- CSI of critical post = 1.08

- Horiz. shear stress of post = 6.341E+01

- Allowable horiz. shear of post = 1.137E+02

- Critical truss chord is: 6-8

- CSI of critical chord = 0.97

- Critical web is: 8-10

- CSI of critical web = 8.0E-02

- Horiz. eave displacement = 1.105E+00

6. Press <return> and the screen displays

**Do you want the reasoning explained
during execution?>>(y/n)**

```

--> Results of consultation

  Because:
    shear force of metal skin = less than allowable

  We conclude that:
    Advice = gauge of skin is okay, cf 100

  Because:
    CSI of critical post = 1.08

  We conclude that:
    Advice = increase bending stress by 10%

  -
  -
  -

          (advice are
          listed here)

          ****End of consultation****

```

7. Enter **y** or **n** depending upon your choice. If **y** is entered, the screen displays the reasoning for the suggested changes to be made on the fn.MCI file. Next, the screen displays (also when **n** is entered)

The program then makes the changes based on the advice and fn.MCI is modified. Next, the screen displays

Creating a new input file for METCLAD...

The above procedure creates a new file called fn.ESO. Now, the program compares fn.MCI and the newly created file fn.ESO. If these two files are the same, the program stops. If not, the fn.MCI and fn.MCO files are erased from the disk and the fn.ESO file is renamed internally to fn.MCI. Next, METCLAD is again executed using this new file and the screen displays

Executing METCLAD...
Iteration no. = 2

and steps 6 and 7 are repeated. A maximum of 50 iterations can be made. After a successful execution, the following files are stored in the disk:

- fn.MCI
- fn.ESO
- fn.MCO

The first two files should be the same (the reason the program stopped iteration), and the third file is the last output of METCLAD. The order of execution of the files is shown in Fig. 2.

When 50 iterations are completed and if fn.MCI is not yet the same as fn.ESO, fn.MCI is stored in the disk and

fn.ESO and fn.MCO are erased. The fn.MCI file is the last file to be updated during the iterative process.

SUMMARY AND CONCLUSIONS

A procedure was developed to optimize designs of metal-clad post-frame rectangular buildings using a knowledge based expert system. The primary objective of the study was to develop a procedure to extend the solution obtained from a conventional algorithmic CAD program to be used by the knowledge base of an expert system. Solving the problem of incompatibility between CAD and expert system programs allows information to be shared between them.

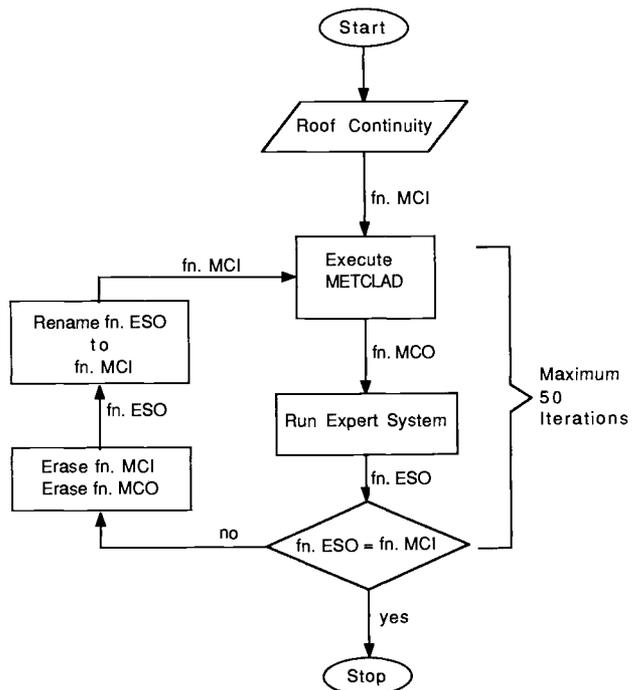


Fig. 2—Sequence of execution.

Met-X-PERT takes the output of the frame analysis program, METCLAD, and uses it as a fact base. The system analyzes the output based on the rules specified. Changes are triggered by the inference engine to modify the input file of METCLAD. METCLAD is then executed again using the modified input file. This sequence of operations continues until all the specified limiting factors for design are met and an "optimum" design is simulated. The advice and actions taken during the process are displayed on the screen.

The program is not intended for practical application for every design is unique and it is difficult to incorporate all requirements, assumptions, and judgements associated with design.

References

1. Adeli, H. 1986. Artificial intelligence in structural engineering. *Engineering Analysis* 3(1):154-160.
2. Adeli, H. and J. Fiedorek. 1986. Microcomputer-aided design and drafting of moment-resisting connections in steel buildings. *Microcomputers in Civil Engineering* 1(1):32-44.
3. Adeli, H. and M.M. Al-Rijleh. 1987. Computer-aided design of trusses using Turbo Pascal. *Microcomputers in Civil Engineering* 2(2):101-116.
4. Alty, J.L. and M.J. Coombs. 1984. *Expert Systems Concepts and Examples*. Manchester, England: NCC Publ.
5. Gebremedhin, K.G. 1986. SOLVER - An interactive structures analyzer for microcomputers. *Transactions of the ASAE* 29(1):10-13.
6. Gebremedhin, K.G. and J.V. Bartlett. 1987. A data preprocessor for structural analysis and design programs. *Journal of Microcomputers in Civil Engineering* 2(2):135-145.
7. Gebremedhin, K.G. 1988a. Computerized system for diaphragm design of metal-clad post frame buildings, *Transactions of the ASAE* 31(1):215-220.
8. Gebremedhin, K.G. 1988b. Expert system development to optimize design of metal-clad post-frame buildings. In *Proceedings 6th National Conference on Microcomputers in Civil Engineers*. ed. W.E. Carroll, 132-138. Nov. 9-11, Orlando, FL.
9. Hayes-Roth, F., D.A. Waterman and D.B. Lenat. 1983. *Building Expert Systems*. Reading, MA: Addison-Wesley Publishing Co.
10. National Forest Products Association. 1986. *National design specification for wood construction*. National Forest Products Association, Washington, D.C.
11. Negoita, C.V. 1984. *Expert Systems and Fuzzy Systems*. Menlo-Park, CA: Benjamin-Cummings Publ. Co. Inc.
12. Suddath, S.K. and R.W. Wolfe. 1984. *Purdue Plane Structures Analyzer II (PPSA II): A computerized wood engineering system*. USDA Forest Products Laboratory, Madison, WI.
13. Waterman, D.A. 1986. *A Guide to Expert Systems*. Reading, MS: Addison-Wesley Publishing Co.