1986

Problem solving using expert system techniques

Mary Lou. Maher
Carnegie Mellon University

Carnegie Mellon University.Engineering Design Research Center.

Problem Solving Using Expert System Techniques

by

M. L. Maher

EDRC-12-02-86

September 1986

Problem Solving Using Expert System Techniques

Mary Lou Maher, A.M.ASCE[1]

1. Introduction

Computers have traditionally been used to solve engineering problems that are formalized and analytical in nature. Using conventional programming techniques, a list of sequentially executable statements must be formulated before the computer can solve the problem. This requirement for explicit formalization of the problem into detailed, sequential statements has restricted the use of the computer to problems that have solutions that are well understood. The desire to use the computer to aid in the solution of engineering problems that are less formalized or understood has led to the recent interest in expert system techniques. Problems are solved in expert systems by a set of rules and/or procedures whose execution order depends on the problem solving strategy utilized.

There are many alternative problem solving strategies that can be implemented using expert system tools and techniques. However, there are basically two approaches to problem solving currently used in expert systems: the derivation approach and the formation approach. The derivation approach involves deriving a solution that is most appropriate for the problem at hand from a list of predefined solutions stored in the knowledge base of the expert system. The formation approach involves forming a solution from the eligible solution components stored in the knowledge base. Depending on the complexity of the problem being solved, an expert system may use one or both of the approaches described above.

This paper begins with a description of a subset of the problem solving strategies used in expert systems and how these strategies support the derivation or formation approach. This is followed by the description of a structural design problem and its implementation using first a formation approach and then a derivation approach. Finally, some comments and conclusions are provided.

2. Problem Solving Strategies

Problem solving involves, the search for a solution. The search begins at an initial state of known facts and conditions and ends at a goal state. The solution path consists of all states that lead from the initial sate to the goal state. In a formation approach to problem solving the known facts and conditions are combined to form a goal state. In a derivation approach, the known facts and conditions are used to derive the most appropriate goal state.

Domain independent problem solving strategies are commonly referred to as weak methods and may lead to combinatorial explosions due to a potential lack of focus. Expert systems can be considered strong problem solvers since they employ domain knowledge in the solution strategy. In this section a number of problem solving strategies currently used in expert systems are briefly presented and discussed in light of their potential for implementing a

[1] Assistant Professor of Civil Engineering, Carnegie-Mellon University Pittsburgh, PA 15213

formation or derivation approach. More detailed descriptions of a number of problem solving strategies can be found in [Nillson 80. Rich 83. Stefik 77], and a review of problem solving strategies appropriate for engineering design in [Maher 84a],

The following strategies are appropriate for the implementation of a derivation approach: forward chaining, backward chaining, and mixed initiative. These strategies require that the goal states represent the potential solutions and the initial state represent the input data. The use of these strategies require the development of an inference network representing the connections between initial states and goal states, as illustrated in Figure 1. The advantage to using one of these strategies is that they are currently implemented in a variety of expert system tools so that the development process involves defining, testing, and revising an inference network.

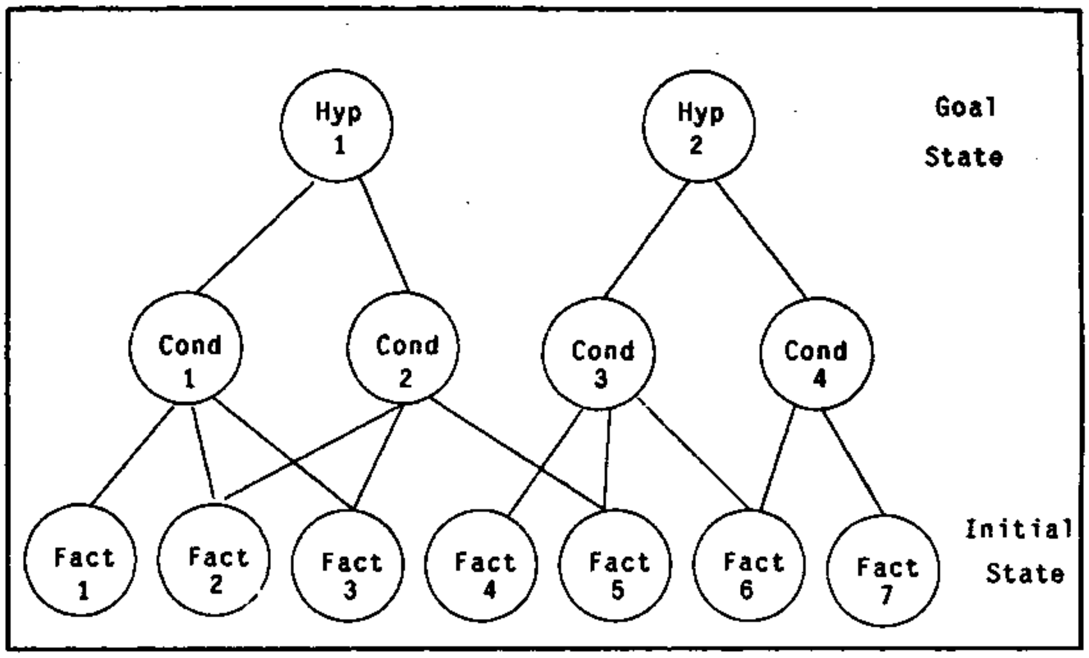


Figure 1: Inference Network For A Derivation Problem

Forward Chaining. A system uses a forward chaining (bottom-up, data-driven, antecedent-driven are all equivalent to forward chaining) strategy if it works from an initial state of known facts to a goal state. The input to such a system includes the value of all facts that the system has in its knowledge base. The main drawback of this strategy is that it is extremely wasteful to require as input data all the possible facts for air conditions; in many circumstances all possible facts are not known or relevant. This strategy is useful in situations where there are a large number of hypotheses and few input data.

Backward Chaining. A system uses a backward chaining (also referred to as top-down, goal-driven, hypothesis-driven and consequent-driven) strategy if it tries to support a goal state or hypothesis by checking known facts. If the known facts do not support the hypothesis then

Problem Solving Using Expert System Techniques

Mary Lou Maher

ABSTRACT

There are many alternative problem solving strategies that can be implemented using expert system tools and techniques. However, there are basically two approaches to problem solving currently used in expert systems: the derivation approach and the formation approach. This paper begins with a description of a subset of the problem solving strategies used in expert systems and how these strategies support the derivation or formation approach. This is followed by the description of a structural design problem and its implementation using first a formation approach and then a derivation approach.

the preconditions that are needed for the hypothesis are set up as subgoals. This process continues until the original hypothesis is either supported or not supported by known facts. The system may then pursue the validity of another hypothesis in the knowledge base. The order in which the hypotheses are pursued is predefined.

Mixed Initiative. A system uses a mixed initiative strategy when it combines the forward chaining and backward chaining strategies. The system starts with the initial state of known facts to assign a probability to each of the potential goal states. The system then tries to support the goal state with the highest probability by setting up subgoals and requesting additional information from the user if necessary. In the mixed initiative strategy the order in which the hypotheses are checked depends on the problem at hand. The advantage to this strategy is that the user only supplies the data relevant to the problem at hand and not all possible data values.

The problem solving strategies appropriate for implementing a formation approach are: problem reduction, plan-generate-test, and agenda control. These strategies may be supplemented with the concepts of hierarchical planning and least commitment, backtracking, and constraint handling techniques. The development of an expert system using one of these strategies requires the definition of the components of the solution and a description of how the components can be combined. An illustration of the unconnected graph of components is shown in Figure 2. The solution is not completely defined by a goal state, but requires that the solution path be known also. The disadvantage to using one of these strategies is the lack of a standard implementation or expert system tool that employs a strategy appropriate for the formation approach. These strategies are typically implemented using a lower level language such as Lisp.

Problem Reduction. Problem reduction involves factoring problems into smaller subproblems. The problem is represented as an AND-OR graph. An AND node consists of arcs pointing to a number of successor nodes, all of which must be solved for the AND node to be true (or solved). For an OR node, it is sufficient for one of the successor nodes to be solved; an OR node indicates that a number of alternate solutions exist for the problem.

Plan-Generate-Test. The generate-and-test strategy in its pure form generates all possible solutions from components in the knowledge base and tests each solution until it finds a solution that satisfies the goal condition. The plan-generate-test sequence restricts the number of possible solutions generated by an early pruning of inconsistent solutions. The pruning is achieved by the planning stage, where the data is interpreted and constraints are evaluated; these constraints eliminate solutions that are inconsistent.

Agenda Control. The agenda control strategy involves assigning a priority rating to each task in the agenda. The task with the highest priority is performed first. A list of justifications and a method of determining the priority rating is associated with each task. This type of control can be used for complex tasks that require focusing attention on certain parts of the problem. Agendas can also be used in systems that require several independent sources of expertise to communicate with each other.

Hierarchical Planning & Least Commitment. The concept of hierarchical planning involves the development of a plan at successive levels of abstraction. For example, in the design of a complex system the design space is divided into a set of levels, where the higher levels are abstractions of the details at lower levels; the problem is hierarchically decomposed into

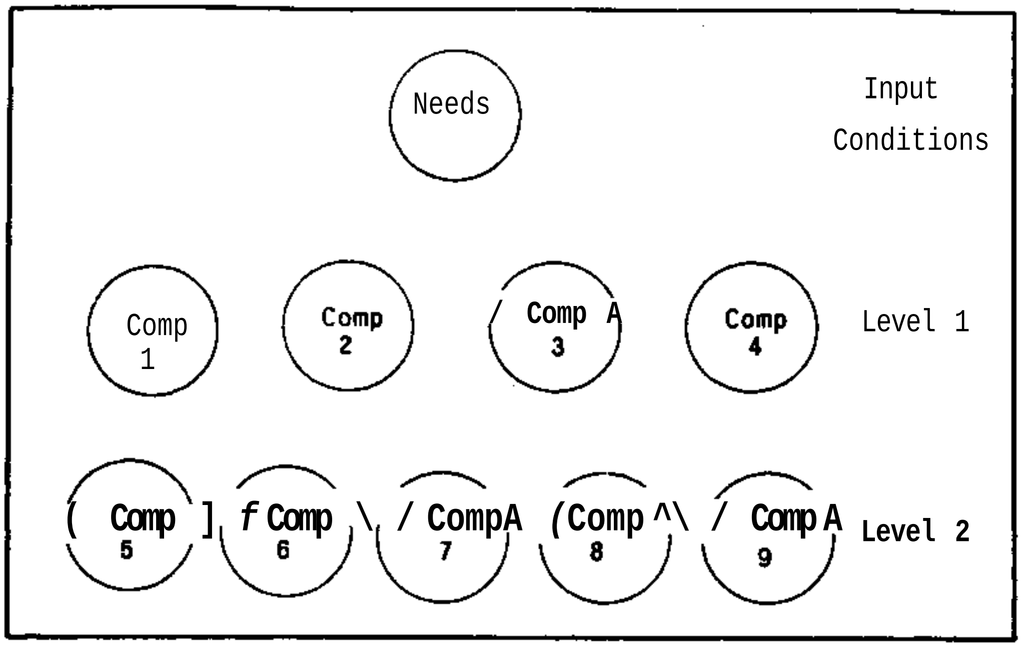


Figure 2: Unconnected Graph For A Formation Problem

loosely coupled subsystems. The least commitment principal involves deferring the assignment of values to variables until more information about the problem space is available. Least commitment is appropriate when a number of solutions exist for each subsystem and the solution of one subsystem depends on decisions made in the solution of another subsystem.

Backtracking. In backtracking the problem solver backs up to a previous level in the solution process if no solution is found along the current path. Backtracking is necessary when a number of alternate approaches to a problem solution exist. Backtracking is provided in some languages, such as PROLOG [Clocksin 81], but must be programmed in most languages. Backtracking, in its pure form, poses a number of difficulties. To provide an efficient way of backtracking from wrong guesses, Stallman and Sussman [Stallman *77]* developed the concept of dependency-directed backtracking (DDB). In DDB, a record of all deduced facts, their antecedent facts along with their justifications are maintained; this information is known as dependency records. The records are used to drive the backtracking process when a contradiction occurs. This concept requires a lot of bookkeeping, but the additional bookkeeeping may be used for explanation purposes as well.

Constraint Handling. The interaction between subsystems can be handled by constraint satisfaction techniques. Constraint satisfaction techniques involve the determination of problem states that satisfy a given set of constraints. Stefik [Stefik 80] proposes three operations on constraints:

1. ***Constraint formulation* is the operation of adding new constraints representing**

restrictions on variable bindings. Typically, the constraints contain increasing detail as the solution progresses.

2. *Constraint propagation* is the operation of combining old constraints to form new constraints. This operation handles interactions between subproblems through the reformulation of constraints from different subproblems.

3. *Constraint satisfaction* is the operation of finding values for variables so that the constraints on these variables are satisfied.

3. Example: Configuring Alternative Lateral Load Resisting Systems

The example selected to illustrate the flexibility of the expert system approach to problem solving is a structural design problem. The ideas used to develop the problem where taken from two expert systems developed at Carnegie-Mellon University: HI-RISE [Maher 84b] and LOW-RISE [Camacho 85]. These expert systems represent an effort to use expert system techniques to study the preliminary structural design process. HI-RISE is an expert system that configures and evaluates alternative structural systems for high rise buildings with limited capabilities for spatial reasoning. LOW-RISE configures and evaluates alternative structural systems for low rise industrial buildings giving some consideration to spatial reasoning. HI-RISE uses a formation approach; the alternative structural systems are formulated by combining and proportioning structural subsystems and components. LOW-RISE uses a derivation approach; the structural components and subsystems are combined and stored in the knowledge base as alternative configurations. The example in this paper is a subset of the design performed by HI-RISE; the definition of feasible alternative lateral load resisting systems. The design will be implemented first using the formation approach that HI-RISE uses and then using the derivation approach that LOW-RISE uses.

3.1 Problem Statement

The problem is to define alternative feasible structural configurations for the lateral load resisting system for a given building. The input to the problem is a three dimensional grid representing the potential locations of structural subsystems and components, as shown in Figure 3. The output is a list of feasible structural alternatives.

The problem statement is further refined by defining the structural subsystems and components that are to be considered. The components are decomposed into hierarchical levels of alternatives. The three levels used for this example are 3D subsystems, 2D subsystems, and lateral system material. The alternatives in each level are shown in Figure 4.

The 3D subsystems are: tube, a three dimensional moment resisting structure placed on the perimeter of the building; core, a three dimensional moment resisting structure placed around an internal core; and orthogonal 2D, two dimensional moment resisting structures placed in orthogonal directions. The 2D subsystems are braced frame, rigid frame, and shear wall. The lateral system materials considered are steel and reinforced concrete.

Heuristics are used to determine whether the combination of alternative subsystems under consideration represents a feasible system for the given problem. Some heuristics are based on material subsystem compatibility. For example, the selections of steel as the material and shear wall as the 2D subsystem is not a feasible combination. Other heuristics are based on

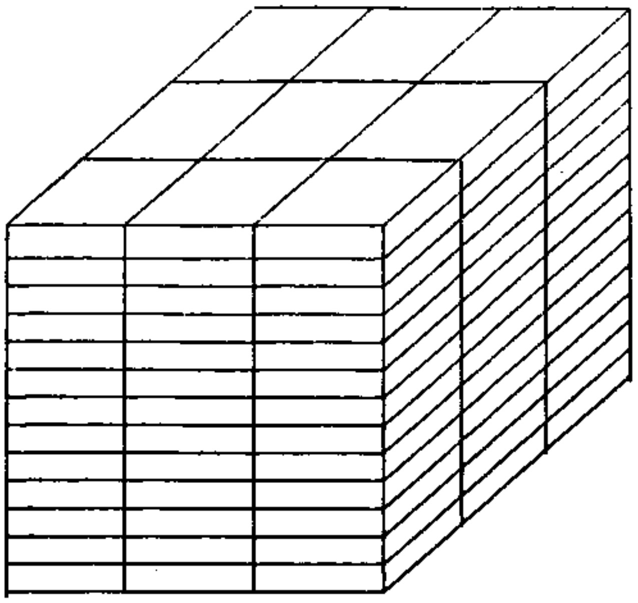


Figure 3: Input To Lateral System Configuration Problem

LEVEL:	RANGE:
3D Subsystem	tube, core, orthogonal 2D
2D Subsystem	braced frame, rigid frame, shear wall
Lateral Material	steel, reinforced concrete

Figure 4: Subsystem Levels For Lateral System Configuration Problem

experience derived from previous building designs. For example, the selection of tube as the 3D subsystem **for** a building less than 40 stories is not a structurally viable solution.

3.2 Formation Approach

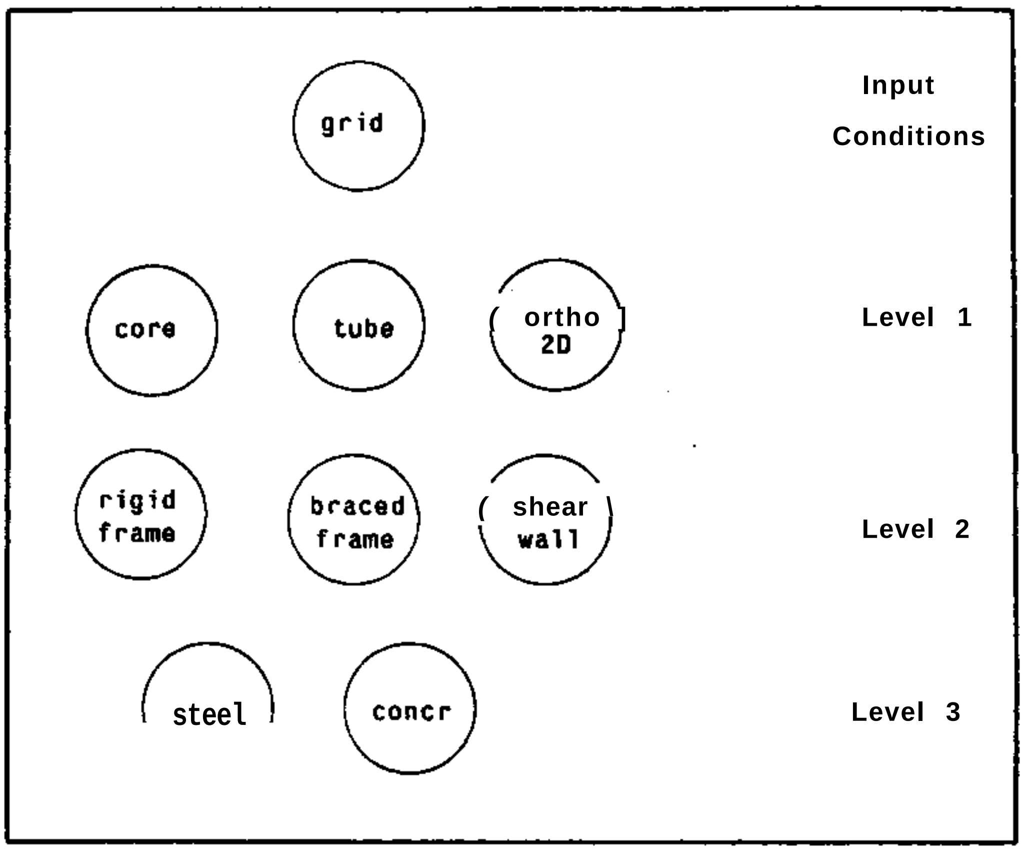


Figure 5: Unconnected Graph For Lateral System Design

The formation approach to the design problem involves the development of an unconnected graph to represent the hierarchical decomposition of the component alternatives, as shown in Figure 5. A generate and test strategy is used to define the feasible alternatives. The components are combined using a depth-first processing of the hierarchy, checking constraints at each level in the hierarchy.

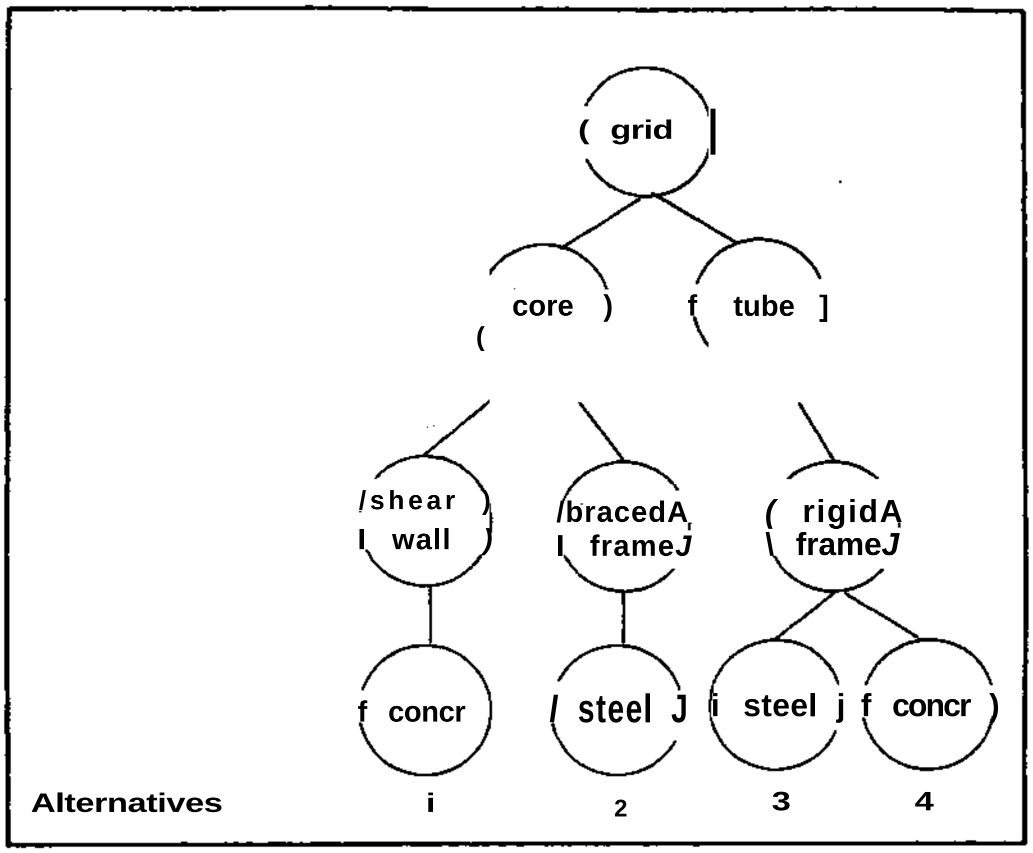
The constraints associated with each level represent design heuristics. For example, some constraints associated with the 30 subsystem level are given below.

```
IF number of stories < 40, AND
   3D subsystem = tube
THEN eliminate alternative.

IF number of stories < 40, AND
   3D subsystem = core
```

```
THEN eliminate alternative.

IP number of stories > 40, AND
   3D subsystem = orthogonal 2D
THEN eliminate alternative.
```



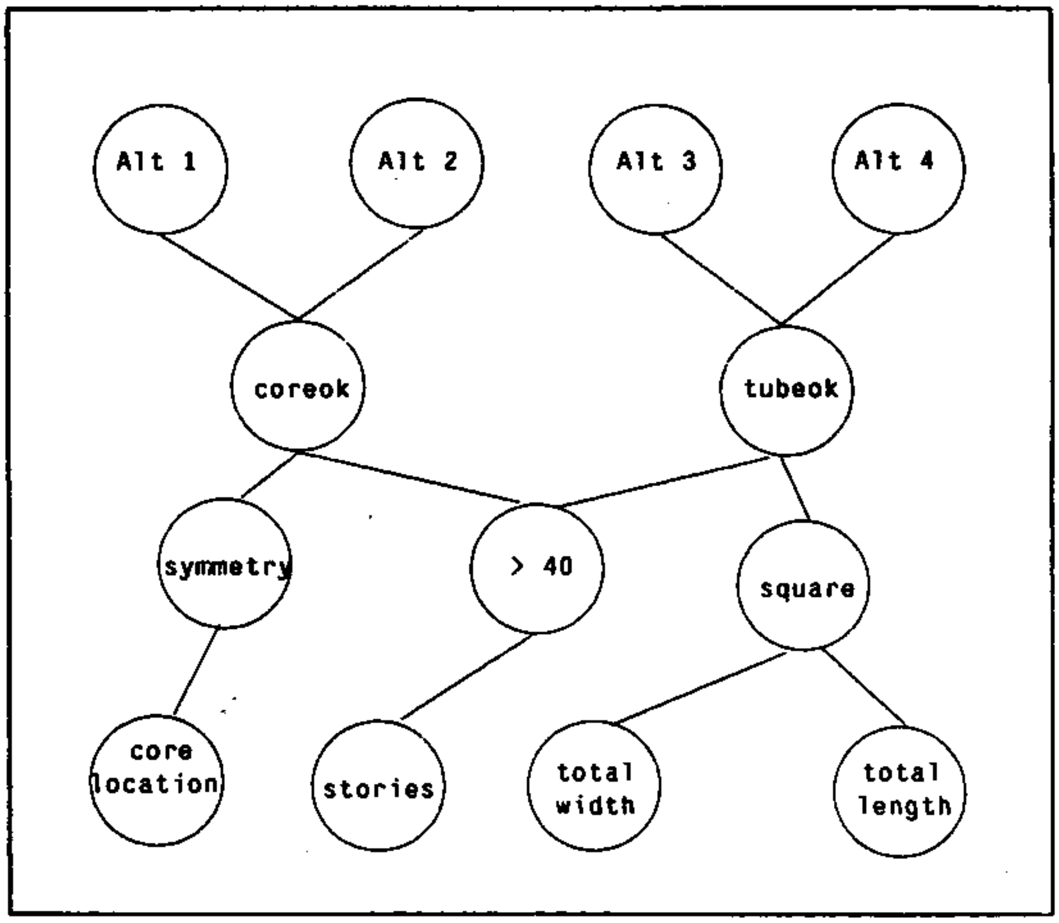
Figu re 6: Tree Of Alternative Configurations

The design process proceeds by selecting one alternative from each level of the hierarchy and checking the elimination constraints. If the alternative is eliminated, another item is selected from the same level. If the alternative is not eliminated, an item is selected from the next level. The result is a tree of alternative configurations, representing the solution path, as shown in Figure 6. The figure illustrates the following alternatives: alternative 1 is a combination of a core, a shear wall, and reinforced concrete; alternative 2 is a combination of a core, a braced frame, and steel; alternative 3 is a combination of a tube, a rigid frame, and steel; and alternative 4 is a combination of a tube, a rigid frame, and reinforced concrete.

3.3 Derivation Approach

The derivation approach to the design problem involves the development of an inference network to represent the connections between feasible solutions and input data, as shown in Figure 7. The feasible solutions represent predefined alternative configurations. For example, **alternative 1 represents a reinforced concrete shear wall** around the core, alternative 2 **represents a steel braced frame around the core, alternative 3 represents a steel rigid frame tube, and alternative 4 represents** a **reinforced concrete rigid frame tube.**

The heuristics that were represented as elimination constraints in the formation approach are reresented as decisions in the inference network in the derivation approach. The design can be implemented using a forward chaining, backward chaining, or mixed initiative strategy. Using the backward chaining strategy, the first alternative is pursued by traversing the network to check if the input data can support the alternative. The other alternatives are pursued in turn. The solution is represented by the alternatives that can be supported by the input data.



Figu re 7: Inference Network For Lateral System Design

4. Summary and Conclusion

This paper presents a review of the most common problem solving strategies used in the development of expert systems in light of implementing either a derivation or formation approach to problem solving. The difference between a derivation and formation approach is illustrated by the implementation of a design problem using both approaches. It is important to note that the flexibility in using either a derivation or formation approach for a design problem is possible only when the design problem is simple. A complex design problem, involving design of multiple, interacting subsystems, cannot be easily implemented using a pure derivation approach.

The author's experience in developing expert system for structural design has been that the initial problem solving approach appropriate for design problems is a formation approach, as this approach is similar to the approach that humans use. Only after the definition of the relevant components and constraints on their combination using a formation approach is it possible to recast the solution strategy into a derivation approach.

The advantage to using expert system techniques in solving ill-structured problems, such as design, is that the problem solving strategy is transparent. A transparent problem solving approach is a definite advantage when the problem solving process may be subject to change. The use of expert system techniques allow changes to be readily recognized and relatively easy to implement. And as is shown in the example in this paper, using a different approach for the same problem can give new insight into the solution of the **problem.**

5. References

[Camacho 85] Camacho, Godofredo, *LOW-RISE: An Expert System For Structural Planning And Design Of Industrial Buildings,* unpublished Master's Thesis. Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh. PA 15213, 1985.

[Clocksin81] Clocksin, W. F. and Mellish, C. S., *Programming in Prolog,* Springer-Verlag. Berlin Heidelberg New York, 1981.

[Maher84a] Maher, M.L. , Sriram, D., Fenves, S.J., 'Tools and Techniques for Knowledge Based Expert Systems for Engineering Design," *Advances m EngineeringSoftware,* **1984.**

[Maher84b] Maher, M.L., *HI-RISE: A Knowledge-Based Expert System For The Preliminary Structural Design Of High Rise Buildings,* unpublished Ph.D Dissertation, Department of Civil Engineering, Carnegie-Mellon University **1984.**

[Nillson80] **Nillson,** N. J., *Principles of Artificial Intelligence,* Tioga Publishing Company, Palo Alto, California, 1980.

[Rich 83] Rich, E., *Artificial Intelligence,* McGraw Hill, 1983.

[Stallman77] Stallman, R., and Sussman, G. J., "Forward Reasoning and Dependency directed Backtracking in a System for Computer-Aided Circuit Analysis." *Artificial Intelligence,* Vol. 9, pp. 135-196, 1977.

[Stehk 77] **Stefik, M. and** Martin. **N..** ***A*** *Revinw of Knowledge Based Problem SoMn$_n$* as *a Bas,s for ,-, Genetics Experiment Designing System,* **Technical Report STAN-CS-77-596, Computer Science Department. Stanford University, March 1977.**

[Stefik 80] **Stefik, M.,** *Planning With Constraints,* **Technical Report STAN-CS-80-784 Computer Science Department. Stanford University. January 1980.**

List of Figures