Pergamon

Mathl. Comput. Modelling Vol. 20, No. 9, pp. 75–87, 1994
Copyright©1994 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0895-7177/94 $7.00 + 0.00

0895-7177(94)00164-2

The Problem of Asking the Minimum Number of Questions in Horn Clause Systems

E. TRIANTAPHYLLOU
Department of Industrial Engineering, Louisiana State University
3134 CEBA Building, Baton Rouge, LA 70803-6409, U.S.A.
IETRIAN@LSUVM.SNCC.LSU.EDU

J.-C. WANG
Department of Business and Economics, Missouri Western State College
St. Joseph, MO 64507-2294, U.S.A.
WANG@ACAD.MWSC.EDU

(Received January 1994; revised and accepted July 1994)

Abstract—This paper deals with the issue of question asking strategies in expert systems with Horn clause rule bases. A good strategy would ask as few questions as possible to reach a conclusion. This paper describes the development of an efficient and effective heuristic approach which is an extension of the strategy developed by Wang and Vande Vate [1]. Similar to the original strategy, the proposed approach is also organized into two phases. In the first phase, a set of candidate questions is formed as in the original strategy. In the second phase, a question is selected from the previous set. A new question selection rule is used in the proposed strategy. This rule is optimal, given a set of candidate questions. Furthermore, computational results indicate that the new question asking strategy is a highly effective and efficient practical approach.

Keywords—Question asking strategies, Horn clauses, Inference engine, Backward chaining, Expert systems, Set cover problem, Heuristic approaches.

1. INTRODUCTION

In a rule based expert system, the user sets a final goal to be proved or he may supply the system with a set of initial data (or facts), and then ask the system to derive all possible conclusions. If the given set of data is insufficient to reach the final conclusion, then the inference engine of the system asks the user (or retrieves the information from sensors) to supply additional information. One of the main objectives of a successful expert system is to be able to reach a conclusion (goal) by asking a small number of questions. This objective becomes more urgent when the rule base involves a vast number of rules.

When the number of rules in the system is very large, the number of questions posed by the system may become too large. If the number of questions increases, so does the possibility of entering wrong answers to some of these questions. Furthermore, even if the user provides the system with only correct answers, still a very large number of questions may make certain applications infeasible because it would take too much time to reach the final conclusion. The above observations indicate that one of the most crucial issues in designing an efficient and effective inference engine is the use of a question asking strategy which yields a small (hopefully near minimum) number of questions. **This paper presents an efficient and effective question asking strategy for rule based expert systems which use Horn clauses.**

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

A problem related to the need of using the minimum number of questions is examined in designing fault diagnostic systems of electrical circuits. The behavior discrepancy of an electrical circuit between the observed and predicted performances indicates malfunctioning of the circuit. When behavior discrepancies are observed, one wants to identify the causes of the discrepancies. In [2], a general minimum entropy technique is presented which examines the differences between a model of an artifact and the artifact itself.

That technique was used to select the best measurement (test) to make next. The required data for this technique are failure probabilities for the components of the artifact. In a later development, de Kleer [3] proposed an efficient measurement selection strategy which uses the assumption that all components of a device fail independently with very small and equal probability. This approach can also be used to test the *integrity* of a Horn system (i.e., to examine whether there are incorrect rules). This is possible because a Horn system can also be approached as a logic circuit with "AND" and "OR" gates.

The problem examined in this paper is different from the one examined in building efficient diagnostic systems. In this paper, all rules are *assumed to be correct* and there is no issue of questioning the integrity of the rule base. The main concern is the development of an efficient and effective question asking strategy for expert systems which use Horn clauses as their rules.

The present paper concentrates on Horn clause systems because these systems have been widely accepted in many real life applications of expert systems technology. In particular, Jeroslow [4, p. 97] states: "recall that only Horn clauses are permitted in expert systems." Hooker [5] points to the same issue by observing that Horn clauses have been used in such systems such as MYCIN [6,7], CADUCEUS [7], and XCON (R1) [8,9], and the logical programming language PROLOG [10].

Usually, expert systems use backward and/or forward chaining for the inference process. Typically these systems select goals and questions contingently. That is, a tentative top level conclusion is chosen arbitrarily and then the system poses questions about any unconfirmed observable assertions encountered in the process of backward chaining [6].

Some of the early applications of experts systems consider the issue of question asking strategies. For instance, the system EXPERT, described in [11], uses pre-ordered lists of rules and questions. Also the system PROSPECT [12], which tries to determine the class of ore deposits from geological survey data, uses a scoring function in implementing a question asking strategy. Another question asking strategy, called "Alpha-Beta pruning," was introduced by Mellish [13] for acyclic inference nets. In that strategy, irrelevant questions are dropped from consideration.

In a general rule base (i.e., not necessarily a Horn clause base), it is an NP-complete problem to determine which assertions are not true under a given set of data (see, for example, [14,15]). However, for Horn clause systems, deduction can be carried out in linear time [16]. Furthermore, Jeroslow and Wang in [15] showed that one can determine how an assertion is proved in a Horn system by solving an associated dynamic program or a linear program. They also demonstrated that inference in a Horn clause system can be represented by a Leontief flow problem which can provide information on the proof structure. Moreover, the problem of defining a strategy which can reach a given goal by asking the *minimum number* of questions is NP-hard for both the general case and Horn clause systems [1,17].

In [1] the structure of an efficient (it takes log-linear time) question asking strategy is proposed. That approach is based on *two phases*. In the first phase, a set of candidate questions is selected. Each such question refers to a variable with unknown value. This set of questions is such that if all the corresponding variables have true value, then the current goal would be proven true as well. In the second phase, a question from the previous set is *randomly* selected.

In this paper, it is assumed that the probability that a given variable has true value is known *a priori* and that these probabilities are independent from each other. This paper introduces an efficient variable selection rule to be used during the second phase of the question asking strategy developed by Wang and Vande Vate [1]. Suppose that a set of candidate questions has been determined during the first phase of the original question asking strategy described in [1]. Then, according to this rule, we should first ask about the value (true or false) of the variable which is *least likely to be true*. This variable selection rule makes the original strategy complete. Furthermore, computer experiments on seven different strategies indicate that the new approach is a highly effective and efficient practical question asking strategy.

2. SOME BACKGROUND INFORMATION

An *assertion* in a rule base can be viewed as a binary variable with value either true or false. In this paper, the terms assertion and variable are used to denote the same concept. An assertion is *observable* if its value can be obtained directly from the user without applying any rules in the rule base. Otherwise, it is called a *nonobservable* assertion (variable). An observable assertion is called *unconfirmed* if its value has not yet been deduced or given by the user.

Usually, the process of choosing the next question is divided into two steps [11]: goal selection and question selection. In the goal selection step, the system chooses a tentative top level conclusion to pursue. In the question selection step, the system selects an *unconfirmed observable variable* (assertion), whose confirmation will help reach the selected conclusion, and asks a question about it.

A *Horn clause* is a disjunctive clause with all but at most one literal negated. In other words, a Horn clause is an "IF ... THEN ... " rule in which a finite set of positive assertions (also called the antecedent part) implies at most one positive conclusion (i.e., at most one assertion). A group of Horn clauses is called a Horn clause system. The following example presents such a *Horn clause system*.

EXAMPLE 1. Consider the following 6 clauses which are defined on the 9 variables V_i $(i = 1, 2, 3, \ldots, 9)$:

CLAUSE 1:	If (V_3 is TRUE and V_4 is TRUE), then (V_1 is TRUE);
CLAUSE 2:	If (V_5 is TRUE and V_6 is TRUE and V_7 is TRUE), then (V_1 is TRUE);
CLAUSE 3:	If (V_3 is TRUE and V_5 is TRUE and V_7 is TRUE and V_9 is TRUE), then (V_2 is TRUE);
CLAUSE 4:	If (V_8 is TRUE), then (V_5 is TRUE);
CLAUSE 5:	If (V_8 is TRUE), then (V_6 is TRUE);
CLAUSE 6:	If (V_8 is TRUE), then (V_7 is TRUE);

Given these clauses, then it can be easily verified that the variables V_3, V_4, V_8, and V_9 are observable, while V_1, V_2, V_5, V_6, and V_7 are nonobservable variables.

For every Horn clause system it is possible to construct a directed graph associated with it [1]. In this graph, each assertion and each clause correspond to a node. There is an arc from a clause node to an assertion node exactly if the assertion is the conclusion of the clause. There is an arc from an assertion node to a clause node exactly if the assertion is in the antecedent part of the clause. The directed graph which corresponds to the previous example is depicted in Figure 1. This directed graph will also be used in the proposed strategy.

Let V_i be an unconfirmed *nonobservable* variable. Then we define an *unconfirmed observable set (or UOV set) of the variable* V_i as a set of unconfirmed variables such that if all were confirmed true then V_i would be proved true, but if any one were false, then V_i could not be concluded from the others. Let P_i represent the probability that the variable V_i has "true" value. These probabilities may be estimated directly by experts or by analyzing historic data. It is assumed that these probabilities are *independent* from each other. Among all the UOV sets of a goal (and thus, nonobservable) variable in a Horn system, there is one UOV set, say S, such that the product $\prod_{V_i \in S} P_i$ is maximum. This UOV set is called the *most promising set (or MP set)*.

The motivation for this term is based on the observation that the variables in the MP set of a goal (nonobservable) variable have the largest combined probability to prove the goal. The notion of the MP set was introduced in [1] in order to derive a special class of question asking

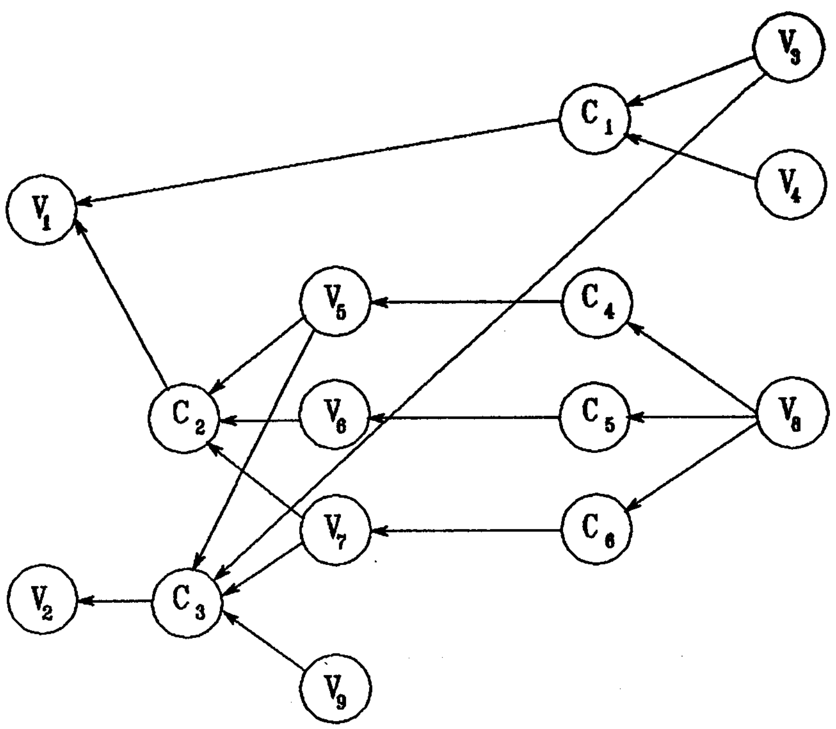

Figure 1. The directed graph for Example 1.

strategies called *sub-effective* strategies. Furthermore, it was proved that to find an MP set of a given goal is an NP-hard problem. Therefore, sub-effective strategies are not computationally efficient since they are NP-hard.

As an alternative to an MP set, they also proposed an approach for calculating a *sub-most-promising set (or SMP set)* of a nonobservable variable. In order to define SMP sets, some additional terminology needs to be introduced. Let V_i and V_k be two variables of some Horn clause system. The *usage of V_i on V_k*, denoted as u_{ik}, is the number of different paths from V_k to V_i in the directed graph associated with the Horn clause system. An SMP set of a variable V_j is defined to be a UOV set of V_j such that the product

$$\prod_{V_i \in S} P_i^{u_{ij}} \text{ is maximum,} \quad (\text{where } u_{ij} \text{ is the usage of the variable } V_i \text{ on } V_j).$$

SMP sets can be viewed as an *approximation* of MP sets. In [1] a labeling algorithm is proposed which can determine an SMP set in log-linear time. Question asking strategies which are based on the use of SMP sets are called *SMP strategies*. The strategy proposed in this paper is an SMP strategy. The above terms and definitions are further illustrated in the following example.

EXAMPLE 2. Consider the Horn clause system presented in Example 1. Recall that the observable variables are: V_3, V_4, V_8 and V_9, while the nonobservable variables are: V_1, V_2, V_5, V_6, and V_7. At this point, *assume* that the probabilities associated with the observable variables are as follows: $P_3 = 0.70$, $P_4 = 0.50$, $P_8 = 0.60$, and $P_9 = 0.40$. It can be easily verified from Figure 1, that for the variable V_1 there are two UOV sets. These are the sets $\{V_3,\ V_4\}$ and $\{V_8\}$. The variable V_2 has only one UOV set: $\{V_3,\ V_8,\ V_9\}$. In the directed graph in Figure 1 there are 3 paths from the nonobservable variable V_1 to the observable variable V_8. These paths are: (V_1–V_5–V_8), (V_1–V_6–V_8), and (V_1–V_7–V_8). Therefore, the usage of V_8 on V_1 is 3. Similarly with the above, it can be shown that the usage of V_4 on V_1 is 1, the usage of V_8 on V_2 is 2, and the usage of V_3 on V_1 is 1.

It should be stated here that, in general, UOV sets and usages cannot be determined efficiently. Next, consider the variable V_1. The corresponding *MP set* is: $\{V_8\}$. This can be easily verified with an exhaustive enumeration in this example because the products which correspond to the two UOV sets $\{V_3,\ V_4\}$ and $\{V_8\}$ have product values equal to 0.35 and 0.60, respectively. Regarding the *SMP set* of the variable V_1, the two related products are as follows:

for the set $\{V_3,\ V_4\}$:	$0.70^1 \times 0.50^1 = 0.35$.
for the set $\{V_8\}$:	$0.60^3 = 0.22$.

Therefore, the corresponding SMP set is $\{V_3,\ V_4\}$ (which is different than the MP set identified above). An efficient way to determine an SMP set is to use a labeling algorithm as follows (see also Figure 2).

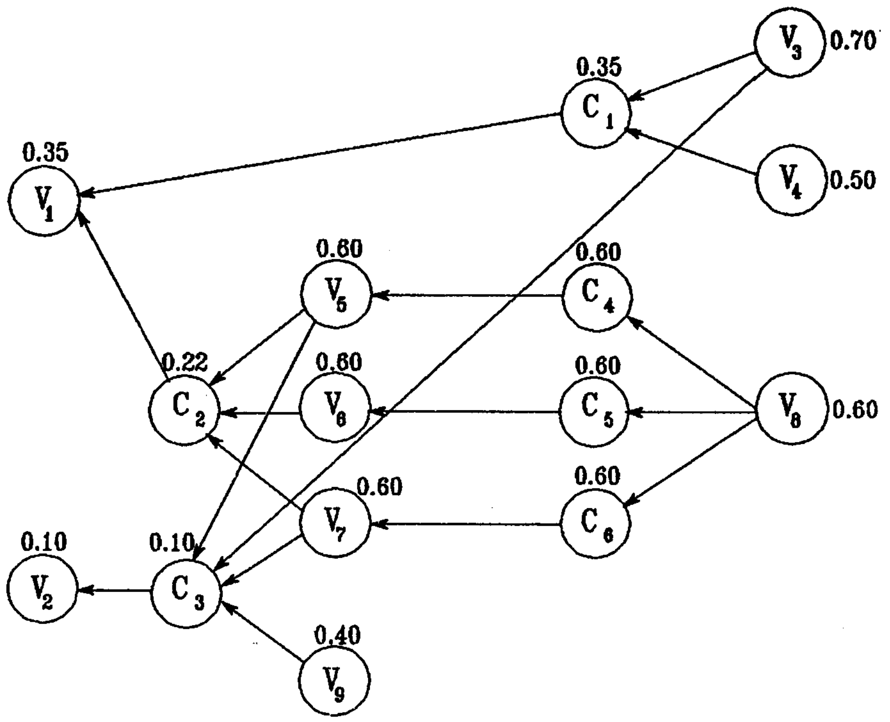

Figure 2. The labels for the graph of Example 1.

According to that algorithm, we start from the nodes which correspond to observable variables. These nodes are given labels equal to the corresponding probabilities. For instance, the node of variable V_3 is given the label 0.70 because $P_3 = 0.70$. Next, the label of a clause node equals to the *product* of the labels of its associated variables. For instance, the node for clause C_1 is given the label 0.35 because $0.70 \times 0.50 = 0.35$. The label of a nonobservable variable is the *maximum* of the labels of the clauses which have that variable as their conclusion. For instance, the label for the variable V_1 is 0.35 because MAX $\{0.35,\ 0.22\} = 0.35$.

The above process needs to be continued until all the nodes in the graph are labeled (see also Figure 2). The label of the goal variable V_1 is 0.35. This label is based on the part of the graph which has as leaves the UOV set $\{V_3,\ V_4\}$. Therefore, the corresponding SMP set is $\{V_3,\ V_4\}$. This is also the SMP set which was identified with the exhaustive enumeration.

In general, the directed graph which corresponds to a Horn clause system may have *cycles*. For that reason, the ASSIGN procedure described in [1] needs to be used. Also, in systems with cycles, it was shown that the labeling algorithm takes log-linear time.

There remain two issues which were unclear in the original SMP strategy:

(1) Given an SMP set, then in what order should we ask the questions from that SMP set?

(2) The original SMP strategy was designed as an efficient (since it requires log-linear time to determine an SMP set) heuristic strategy. How far is it from being a truly optimal strategy?

For the first issue, an optimal rule for asking questions, *given as SMP set*, will be proposed in the next section. Next, this rule will be added to the original SMP strategy. The new strategy will be tested in random computational experiments in Section 4. These tests show that the new SMP strategy is a good approximation of an optimal strategy (i.e., of a strategy which reaches a conclusion by asking the minimum number of questions).

3. ASKING FIRST ABOUT THE LEAST-LIKELY-TRUE VARIABLE

As it was mentioned earlier, the *original SMP* strategy follows a two phase approach. During the first phase, a set of candidate questions (and hence variables) is determined in log-linear time. This is an SMP set of the goal variable. In the second phase, the system randomly selects one of the questions determined in the first phase and asks the user about its value. If the answer is positive (i.e., the corresponding variable has "true" value), then the system randomly selects another question from the same set of candidate questions. However, if the answer is "false," then the first phase is initiated again, and a new set of candidate questions needs to be determined. The following theorem states an optimal rule for selecting the order of the questions from a *given* UOV set.

THEOREM. *If a question in proving a goal V_k is to be selected from a given UOV set of V_k in a Horn clause system, then in order to minimize the expected number of questions within this UOV set, the next question must be about the value of the variable which is least likely to be true compared to the other variables in this UOV set.*

PROOF. Suppose that there are t variables (denoted as V_i, for $i = 1, 2, 3, \ldots, t$) in the given UOV set of the target goal variable V_k. Furthermore, suppose that these variables are selected to be asked in the order of V_1, V_2, V_3, ... , V_t. Let P_i denote the probability that variable V_i (for $i = 1, 2, 3, \ldots, t$) has true value . Then, E, the expected number of questions to ask is:

$$E = 1 + P_1 + P_1P_2 + P_1P_2P_3 + \cdots + \prod_{i=1}^{t-1} P_i.$$

Next, it can easily be shown by contradiction that a necessary condition for E to be minimum is:

$$P_j = \min\{P_j, P_{j+1}, P_{j+2}, \ldots, P_t\}, \qquad \text{for } j = 1, 2, 3, \ldots, t-1.$$

That is, the questions should be asked in ascending order of their corresponding probabilities. This completes the proof of this theorem. ∎

When an SMP set of the final goal variable V_k is determined by the original SMP strategy (in log-linear time) this theorem establishes a rule for the question selection problem encountered in the second phase. According to this rule, the system should first ask a question about the value of the variable with the least probability to be true. The computer experiments in the next section show that when an SMP set is given, then starting with the least-likely-true variable is significantly more efficient than starting randomly or with the most-likely-true variable. Furthermore, the same experiments indicate that sub-effective strategies (which use MP sets) are only *slightly* better than the corresponding SMP strategies.

Since a sub-effective strategy depends on finding an MP set, that is, a set S which maximizes the product $\prod_{i \in S} P_i$ (by solving an NP-hard problem), the test problems *assumed* that *all* the UOV sets of the goal variable were known. Then, the MP and SMP sets can be found easily by performing a simple examination on all the UOV sets of a test problem. These computer experiments are described in more detail in the next section.

4. COMPUTATIONAL EXPERIMENTS

The computational experiments compared seven question asking strategies on 64,000 randomly generated problems. Each problem was assumed, without loss of generality, to have only one potential goal to be proved. To see the generality of this assumption, consider a system in which the variables $V_1, V_2, V_3, \ldots, V_k$ are its potential goals. This system can be converted into an equivalent *single goal* system by adding one more variable, say V_0, and the following k rules (Horn clauses):

$$\text{IF } (V_i \text{ is true}), \text{ THEN } (V_0 \text{ is true}), \qquad \text{for } i = 1, 2, 3, \ldots, k.$$

The test problems were in the form of a group of UOV sets. Since the objective of the experiments was to determine the number of questions required by each question asking strategy, deduction was not necessary. Therefore, the problems did not have to be in the form of "IF ... THEN ... " rules. The test problems were randomly generated with two parameters:

(1) the number of observable variables, and
(2) the number of UOV sets.

The following eight values: 10, 20, 30, ... , 80 were considered for each parameter. Each observable variable was assumed to have probability 0.50 of belonging to a given UOV set (this value was chosen arbitrarily).

For each combination of a number of variables and a number of UOV sets, 1,000 random problems were tested. The probability P_i, that an observable variable V_i is true, was a random number uniformly distributed in the interval [0, 1]. These probabilities are assumed to be independent from each other. However, the observable variable V_i was assigned to a true value if and only if the following condition was satisfied:

RANDOM $< 5.00 \times P_i$,
where: RANDOM was a random variable uniformly distributed in the interval [0,1].

The above step was implemented in order to generate more "true" values than "false" ones (the value 5.00 is arbitrary). In this way, the questioning search was forced to take more steps, and hence, to simulate more challenging situations. Furthermore, the usages, u_{i0}, were random integers uniformly distributed in the interval [1, 11] (this interval was considered arbitrarily).

It should be stated here that in the computer experiments the probability that a given variable belongs to a given UOV set was *independent* (i.e., always equal to 0.50) from its usage value. However, in a real application, this may not always be the case. For instance, it is possible that if a variable occurs many times in a Horn clause system, then one would expect its usages to be high, and the probability that it is in a given UOV set also to be high.

Each random problem was tested according to the following procedure:

Step 1: Select a UOV set according to one of the seven question asking strategies (to be described later).
Step 2: Select to ask a question about an unconfirmed variable in this UOV set.
Step 3: If the answer is "true," go back to Step 2. If the answer is "false," drop all the UOV sets which include this false valued variable and go back to Step 1.

Stopping rules:

Stop: If all of the variables in a UOV set are true. In this case, the goal is proved to be true.
Stop: If no UOV set is left. In this case, the goal cannot be proved to be true or false.

All the examined strategies were comprised of *two main phases*. In the first phase, a UOV set is selected (Step 1). In the second phase, a variable is selected from the previous UOV set

(Step 2). For the case of selecting a UOV set (Step 1), three major scenarios were examined (see also Table 1). Under the *random scenario* (denoted as RND in Table 1), the UOV set was chosen randomly from the available UOV sets. Under the *sub-effective scenario* (denoted as MP in Table 1), the UOV set was an MP (most promising) set of the final goal. That is, by choosing a UOV set S such that the product $\prod_{V_i \in S} P_i$ was maximum. Under the *SMP scenario*, the UOV set was an SMP set. That is, by choosing a UOV set S such that the product $\prod_{V_i \in S} P_i^{u_{io}}$ was maximum.

The question to be asked next, was based on a variable of the UOV set selected in the first step. Three scenarios were also considered in selecting the next variable (Step 2). In the first scenario the variable was *randomly* selected. The second scenario was to select the *most-likely-true (or MLT)* variable. That is, by first asking the variable which is most likely to be true. The third scenario is to select the *least-likely-true (or LLT)* variable. That is, by first asking the variable which is least likely to be true. Besides the previous six strategies, the strategy of selecting a random UOV set with a random variable (denoted as RND-RND) was also considered. These seven strategies are depicted in Table 1. Some of the previous issues are further illustrated in the following example.

Table 1. The seven strategies.

UOV Set Selection Scenarios		Variable Selection Scenarios		
		Random	MLT	LLT
Random	(RND)	RND-RND	RND-MLT	RND-LLT
Sub-effective	(MP)	Not considered	MP-MLT	MP-LLT
SMP		Not considered	SMP-MLT	SMP-LLT

EXAMPLE 3. Consider a test problem which is defined, for instance, in terms of 5 UOV sets and 5 variables (denoted as V_i, $i = 1, 2, 3, 4, 5$). Suppose that the first UOV is: $\{V_1, V_2, V_5\}$, the second UOV set is: $\{V_2, V_3\}$, the third is: $\{V_1, V_2, V_3, V_4\}$, the fourth is: $\{V_4, V_5\}$, and the fifth is: $\{V_1, V_5\}$. Furthermore, suppose that the corresponding usage values are: $u_{10} = 3$, $u_{20} = 11$, $u_{30} = 2$, $u_{40} = 2$, and $u_{50} = 4$. Also, suppose that the probabilities P_i $(i = 1, 2, 3, 4, 5)$ are as follows: $P_1 = 0.30$, $P_2 = 0.95$, $P_3 = 0.10$, $P_4 = 0.50$, and $P_5 = 0.25$. At this point also assume that the following "true"/"false" values have been randomly (according to the procedure described earlier in this section) assigned: V_1 is TRUE, V_2 is TRUE, V_3 is FALSE, V_4 is TRUE, and V_5 is TRUE.

Suppose that the SMP-LLT strategy is to be applied. Given the previous data, it can be easily seen that the SMP set is the second UOV set (the corresponding product takes the value of $0.95^{11} \times 0.10^2 = 5.69 \times 10^{-3}$). Therefore, the set of candidate questions which is generated in Step 1 under the SMP-LLT strategy, is based on the variables $\{V_2, V_3\}$. Since variable V_3 is the least likely to be true, the first question will be about the value of this variable. When this question is asked, *only then it is revealed* to the system that the value of V_3 is "false."

At this point, the SMP-LLT strategy will drop from consideration all the UOV sets which contain variable V_3 and will go back to Step 1. Now there are only three remaining UOV sets. These are the sets: $\{V_1, V_2, V_5\}$, $\{V_4, V_5\}$, and $\{V_1, V_5\}$. It is easy to verify that the new SMP set is the set $\{V_1, V_5\}$. Similarly as above, the system will next ask a question about the value of the variable V_5. The answer now is "true." Therefore, the system will next ask a question about the value of the variable V_1. The answer is again "true." Since no more variables are left in the current SMP set, the questioning process stops, and the final goal is proven to be true after asking a total of 3 questions. ∎

As it can be seen from the way these test problems were considered in this investigation, the MP and SMP sets can be determined by simply examining the values of the corresponding products. In this way, the performance of a sub-effective strategy can be studied *without* having to solve the NP-hard problem needed in finding the required MP set. In other words, the test problems *assumed* that *all* the related UOV sets are known *a priori*. However, in a *real situation*, the UOV sets are *not known a priori* and determining an MP set is an NP-hard problem, while finding the SMP set takes only log-linear time [1].

Table 2. Average number of questions under different strategies.

No. of Observable Variables	Strategy							Lower Bound
	RND-MLT	RND-RND	RND-LLT	SMP-MLT	SMP-LLT	MP-MLT	MP-LLT	
10	4.92	4.49	4.09	1.61	1.59	1.55	1.54	1.26
20	11.36	9.79	8.22	5.06	4.70	4.63	4.45	3.76
30	19.18	15.74	12.17	9.99	8.29	9.12	7.90	6.99
40	28.07	22.23	15.83	16.44	11.83	15.16	11.42	10.30
50	37.65	28.81	18.73	24.40	14.97	22.79	14.56	13.32
60	47.47	35.10	20.63	33.66	17.18	31.71	16.76	15.51
70	57.57	40.99	21.69	43.78	18.67	41.75	18.31	17.07
80	67.45	46.24	21.90	54.08	19.27	52.20	18.97	17.86

The results of these experiments are presented in Table 2 and are also depicted in Figures 3 and 4. Figure 3 illustrates the *average* number of questions asked under each strategy. The horizontal axis depicts the number of observable variables in the UOV sets. The vertical axis depicts the average number of questions asked under a given strategy. The number of UOV sets did not play an important role in these results and thus, it is not depicted in these results.

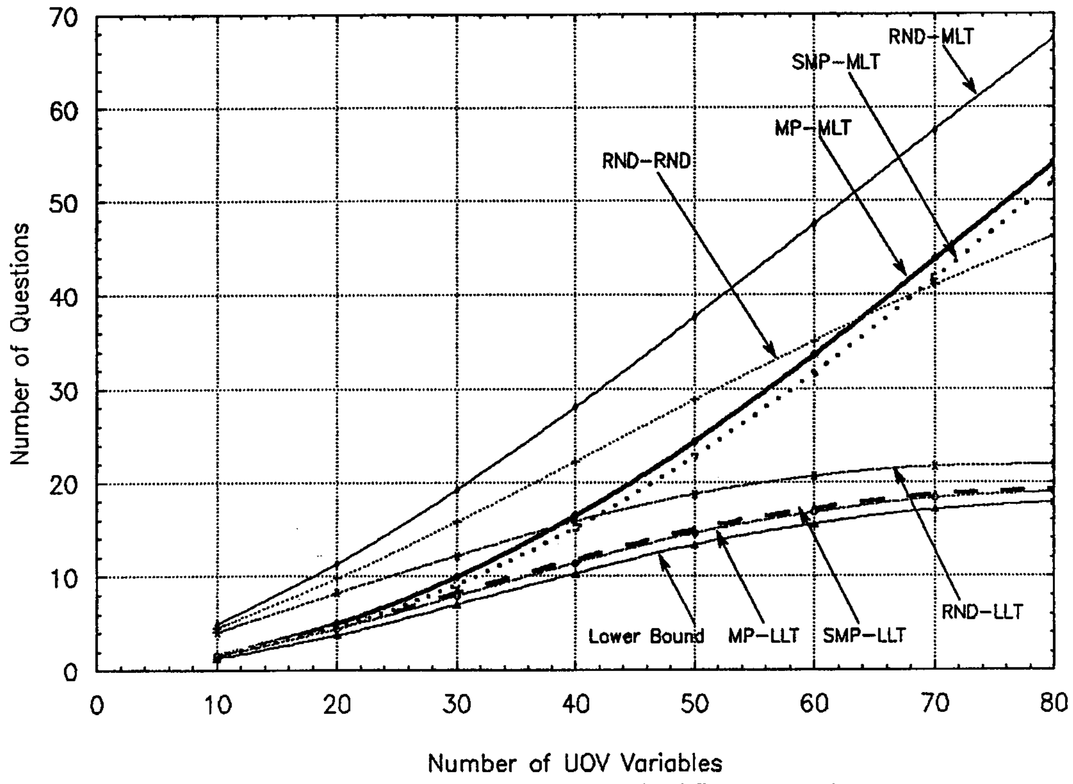

Figure 3. Average number of questions under different strategies.

These figures also present the *lower bound* of questions for each case. Since the *actual* values of the observable variables in the UOV variables were *assumed* to be known for the purpose of these simulated experiments, a lower bound of the number of questions needed to reach the final conclusion can be calculated as follows:

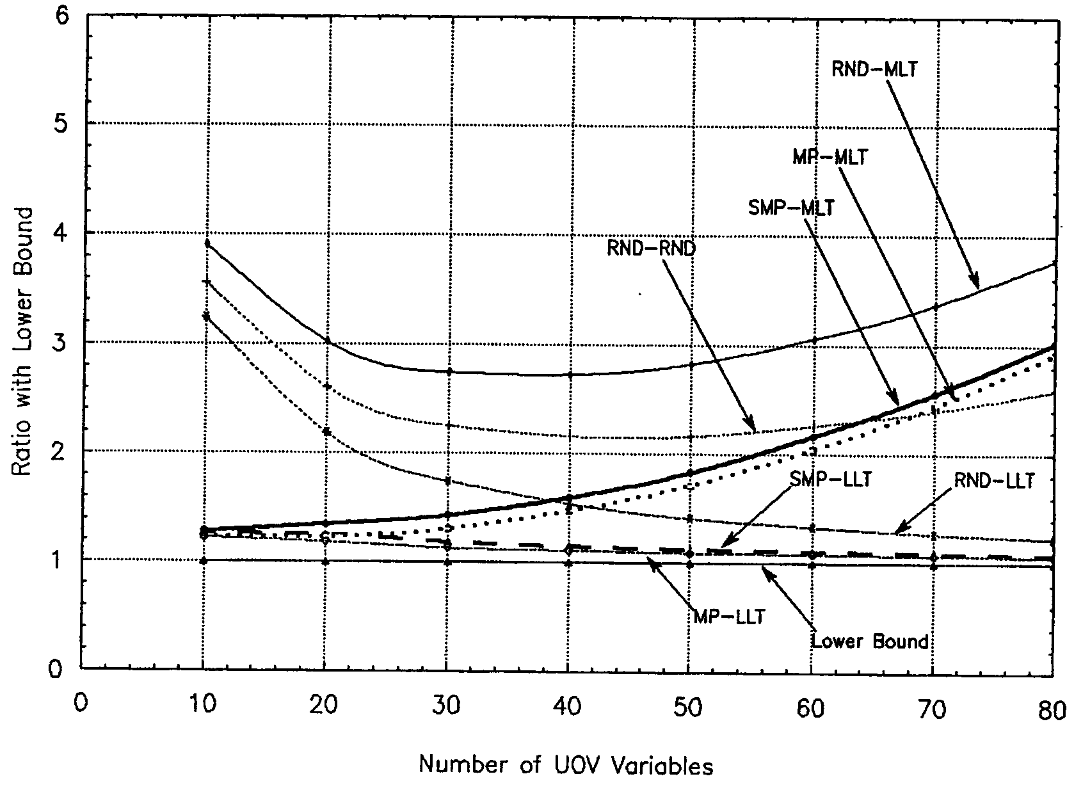

Figure 4. Performance of the strategies relative to the lower bound.

CASE 1. There is at least one UOV set which has all its variables with "true" values. Then, a lower bound on the number of questions which can be asked by any strategy is the *smallest cardinality* of these UOV sets.

CASE 2. The previous case does not hold. That is, all the UOV sets have at least one variable which has "false" value. In this case, a lower bound can be determined by calculating the minimum number of variables with "false" value which cover all the UOV sets. This number is the optimal value of the objective function of the following *set cover problem*:

$$\text{MINIMIZE} \sum_{i=1}^{m} \sum_{X_j \in S_i} X_j$$

$$\text{subject to:} \sum_{X_j \in S_i} X_j \geq 1, \qquad \text{for } i = 1, 2, 3, \ldots, m$$

and all the X_j's are either 0 or 1,

where m is the number of the UOV sets, and the X_j's indicate the variables with "false" value in the UOV sets (denoted here as S_i, for $i = 1, 2, 3, \ldots, m$).

The previous two cases are further explained in the following example.

EXAMPLE 4. For *Case 1*, consider the test problem described in Example 3. The first, fourth, and fifth UOV sets are the only ones which have variables with true value. Moreover, their cardinalities are 3, 2, and 2, respectively. Therefore, no question asking strategy can reach a conclusion by asking *less than 2 questions* (i.e., the value of the smallest cardinality). Therefore, the lower bound is equal to 2.

For *Case 2*, consider the same test problem as above, but now suppose that also the variables V_2 and V_5 have in reality "false" value (i.e., besides the variable V_3). In this case, there is no single UOV set with all its variables having "true" value. Then the corresponding set cover problem takes the following form:

$$
\begin{aligned}
&\text{MINIMIZE } X_2 + X_3 + X_5 \\
&\text{subject to:} \\
&\qquad X_2 + X_5 \geq 1 \text{ (for the 1st UOV set)} \\
&\qquad X_2 + X_3 \geq 1 \text{ (for the 2nd UOV set)} \\
&\qquad X_2 + X_3 \geq 1 \text{ (for the 3rd UOV set)} \\
&\qquad\qquad X_5 \geq 1 \text{ (for the 4th UOV set)} \\
&\qquad\qquad X_5 \geq 1 \text{ (for the 5th UOV set)} \\
&\text{and } X_2,\ X_3,\ X_5 \text{ is either 0 or 1.}
\end{aligned}
$$

In this demonstration, an optimal solution is $X_3 = X_5 = 1$ and $X_2 = 0$. The objective function has value at optimality equal to 2. That is, by asking *at least two* questions, the system can determine that the final goal cannot be proven. Therefore, the lower bound now is equal to 2.

Finally, Figure 4 presents the performance of each strategy *relative* to the lower bound described above. For this reason, the lower bound is represented by a horizontal line with value on the vertical axis equal to 1.00. All the other lines were normalized subject to that line.

5. CONCLUDING REMARKS

The computational results in Table 2 or in Figures 3 and 4 lead to the following observations:

(1) Among the seven examined strategies the best strategy is MP-LLT. Moreover, the strategy SMP-LLT is a close approximation of the MP-LLT strategy.

(2) The worst strategy is RND-MLT. Recall that this strategy randomly selects an UOV set and then asks questions according to the "most-likely-true variable first" (i.e., MLT) rule.

(3) The performance of the strategies which use the "most-likely-true variable first" rule deteriorates (see also Figure 3) as the number of observable variables increases. This, however, is not the case with the strategies which use the "least-likely-true variable first" (i.e., LLT) rule.

(4) Both the MP-LLT and the SMP-LLT strategies are very close to the lower bound (see also Figure 4).

(5) The strategy RND-LLT has very good performance (i.e., it is closer to the strategies MP-LLT and SMP-LLT) when the number of observable variables is high (greater than 70).

Although the computer experiments showed that MP-LLT is the best strategy (in terms of the number of questions asked), this strategy is *not practical* in real life situations. This happens because determining the MP (most promising) set is NP-hard when the problem is given in the form of "IF ... THEN ... " rules. For this reason the best *practical* strategy is the SMP-LLT strategy. In the experiments, the SMP-LLT strategy performed almost as well as the MP-LLT strategy. Moreover, determining an SMP set takes only log-linear time in a real situation where rules are in the "IF ... THEN ... " form. Therefore, *the recommended SMP-LLT strategy is a highly effective and efficient strategy for real life applications.*

The recommended SMP-LLT strategy is an extension of the original SMP strategy which was developed in [1]. In a real life application (i.e., when the rules are Horn clauses of the "IF ... THEN ... " form) the proposed strategy can be applied as follows:

Step 1: Select an SMP set by applying the log-linear time labeling algorithm developed in [1].
Step 2: Select to ask a question about the value of an unconfirmed observable variable in this SMP set by using the "least-likely-true variable first" (LLT) rule.

Step 3: If the answer is "true," go back to Step 2. If the answer is "false," drop all the Horn clauses in the rule base which include this false valued variable in their antecedent parts and go back to Step 1.

Stopping rules:

Stop: If all of the variables in an SMP set are true. In this case, the goal is proved to be true.

Stop: If no SMP set is left. In this case, the goal cannot be proved to be true or false.

The computational experiments also illustrated that the number of UOV sets in a test problem is not important. However, the total number of unconfirmed observable variables in a problem is critical in the performance of the strategies. Another interesting point is to observe that the RND-RND strategy represents the performance under the usual *backward chaining* approach. This strategy has the worst performance for small and large values of the number of unconfirmed observable variables. This can be seen from Figure 4 because the corresponding curve is far away from the lower bound when the number of observable variables is low or very high.

As it was stated earlier in this section the RND-LLT strategy (random set and *"least-likely-true variable first"*) has a very good performance for large numbers of unconfirmed observable variables. This observation suggests that when the number of observable variables is large (i.e., more than 70), then a good idea is to ask about the variable which is least likely to be true without bothering to select an SMP set first. Just any UOV set would be fine. However, it should be kept in mind that, before asking questions, one has to carry out the deduction to see whether the goal can be reached by the currently known facts. An SMP strategy performs deduction and UOV set selection simultaneously [1].

Another important issue is to observe that the small difference between the performance of the proposed SMP-LLT strategy and the lower bound, allows for very little potential for future improvement on that strategy. Future research may be focused on combining the inference process with the question selection strategy, and extending the results on Horn clause systems to more general logical systems. Finally, it should be pointed out that the experimental results generated above are the averages on randomly generated instances. On a particular real world instance, a strategy may perform significantly different from its average.

REFERENCES

1. J. Wang and J. Vande Vate, Question-asking strategies for Horn clause systems, *Annals of Mathematics and Artificial Intelligence* 1, 359–370 (1990).
2. J. de Kleer and B.C. Williams, Diagnosing multiple faults, *Artificial Intelligence* **32**, 97–130 (1987).
3. J. de Kleer, Using crude probability estimates to guide diagnosis, *Artificial Intelligence* **45**, 381–391 (1990).
4. R.G. Jeroslow, Logic-based decision support: Mixed integer model formulation, Special Issue of *The Annals of Discrete Mathematics* (Edited by P.L. Hammer) **40** (1989).
5. J.N. Hooker, A quantitative approach to logical inference, *Decision Support Systems* **4**, 45–69 (1988).
6. B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert Systems, The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, MA, (1984).
7. E.H. Shortliffe, S.G. Axline, B.G. Buchanan, T.C. Merigan and S.N. Cohen, An artificial intelligence program to advice physicians regarding antimicrobial therapy, *Computers and Biomedical Research* **6**, 544–560 (1973).
8. J. Bachant and J. McDermott, R1 revisited: Four years in the trenches, *AI Magazine* **5**, 21–32 (1984).
9. J. McDermott, R1: A rule-based configurer of computer systems, *Artificial Intelligence* **19** (1982).
10. D.H.D. Warren, L.M. Pereira and F. Pereira, PROLOG—The language and its implementation compared with LISP, *Proceedings of the Symposium on Artificial Intelligence and Programming Languages (ACM); SIGPLAN Notices 12; and SIGART Newsletter* **64**, 109–115 (1977).
11. F. Hayes-Roth, D.A. Waterman and D.B. Lenat, *Building Expert Systems*, Addison-Wesley, Reading, MA, (1983).
12. R. Duda, J. Gasching and P. Hart, *Model Design in the PROSPECT Consultant System for Mineral Exploration, Expert Systems in the Micro-electronic Age*, Edinburgh Univ. Press, Great Britain, (1979).
13. C.S. Mellish, Generalized alpha-beta pruning as a guide to expert system question selection, Expert System 85, In *Proceedings of the Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems*, pp. 31–41, Cambridge Univ. Press, Cambridge, (1985).

14. M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman, New York, (1979).
15. R.G. Jeroslow and J. Wang, Dynamic programming, integral polyhedra and Horn clause knowledge bases, *ORSA's Journal on Computing* **1**, 7–19 (1989).
16. W.F. Dowling and J.H. Gallier, Linear time algorithms for testing the satisfiability of Horn formulae, *Journal of Logic Programming* **3** (1), 267–284 (1984).
17. J. Wang, Rule-based expert systems and discrete optimizations, Ph.D. Thesis, College of Management, Georgia Institute of Technology, Atlanta, GA, (1990).