Complexity of the Exact Solution to the Test Sequencing Problem

Wenhao Liu
Management Science and Engineering Dept
Stanford University
Stanford, California 94305
owenliu@stanford.edu

Ross D. Shachter
Management Science and Engineering Dept
Stanford University
Stanford, California 94305
shachter@stanford.edu

Abstract

Consider a doctor choosing a treatment for an uncertain disorder for which there are n costly tests available. Based on the test results observed so far, the doctor can either order another test or proceed to the treatment decision. Although test sequencing is a problem that arises frequently in many decision situations, finding an exact solution is NP-hard with respect to n. In this paper, we analyze the time complexity of classic symmetric and asymmetric formulations, using influence diagrams and decision trees, to the general test sequencing problem, making no assumptions of conditional independence among the test results. We develop an alternative influence diagram formulation that scales better, and show how a decision circuit formulation improves even more on the decision tree solution through recursive coalescence. We prove that this decision circuit formulation achieves the lower bound complexity for any method for the general test sequencing problem that examines the entire policy space. As a result, the problem is tractable for much larger n than has been possible to date.

1 INTRODUCTION

Consider a doctor who must choose a treatment for an uncertain disorder for which there are n costly tests available. After each test is performed, the doctor can order another test or proceed to the treatment decision. Therefore, the doctor would like an optimal strategy for sequencing the tests, taking into account the costs of the tests and all of the test results observed so far. In this paper we examine the computational time complexity of exact solution methods for this problem.

The test sequencing problem is an asymmetric decision problem, where many combinations of uncertain variable states given decision variables have zero probability (Bielza and Shenoy, 1999). The problem arises frequently in many practical decision situations and is of theoretical interest in operations research, machine learning, and the design of experiments. Although finding an exact solution is NP-hard with respect to the number of tests n (Papadimitriou and Tsitsiklis, 1987), the complexity of standard algorithms for this problem has not been analyzed or compared explicitly in the literature. Traditional graphical models used to solve such problems include decision trees (von Neumann and Morgenstern, 1944) and influence diagrams (Howard and Matheson, 1981; Shachter, 1986). Decision trees with limited coalescence (Howard, 1977; Olmsted, 1983) reuse some of the calculations to improve efficiency. There are many other asymmetric decision model representations, primarily designed to improve model formulation, including asymmetric influence diagrams (Smith et al., 1993), valuation networks (Shenoy, 2000), sequential decision diagrams (Covaliu and Oliver, 1995), sequential valuation networks (Demirer and Shenoy, 2006), and unconstrained influence diagrams (Jensen and Vomlelová, 2002). See Bielza et al. (2011) for a review and comparison of such models.

The test sequencing problem involves a decision maker with one key high-stakes decision and a set of information gathering activities. There has been rich and varied research on this problem dating back to the initial work on dynamic programming (Bellman, 1956), but due to the complexity of the problem much subsequent work in this area has focused on designing heuristic and approximate solution methods. Some exact methods feature state variables representing the belief of the decision maker, instead of maintaining all of the observations in the state, such as in Ulu and Smith (2009). Bickel and Smith (2006) consider a test sequencing problem in the context of oil exploration, and advocate "recombining" decision tree models for efficient computation, an example of the recursive coalescence that we focus on in this paper. Despite all of the research in this area, however, there had been no significant improvements in computational complexity for exact solutions to the general test sequencing problem.

Decision circuits, a generalization of decision trees, were developed by Bhattacharjya and Shachter (2007) and Bhattacharjya (2009) to efficiently evaluate influence diagrams with methods similar to arithmetic circuits for Bayesian networks (Darwiche, 2003). Although decision circuits share similar "tree-like" structures with decision trees, especially decision trees with coalescence, they are also as capable of exploiting conditional independence as influence diagrams, and as able to decompose problems as junction trees (Shachter and Peot, 1992; Jensen et al., 1994).

In the next section we formally define the general test sequencing problem, and in the following sections we formulate and analyze the complexity of exact solutions using the symmetric approach of influence diagrams and the asymmetric approaches of decision trees and decision circuits. We show that the decision circuit achieves the lower bound complexity of any algorithm for the problem that examines the entire policy space. We conclude with a comparison of the results and their implications.

2 NOTATION AND FRAMEWORK

In the *general test sequencing problem*, a doctor is treating a patient who has an uncertain disorder D with b possible states and probability distribution $\Pr\{D\}$. The doctor will choose a treatment T_x from among a alternatives in order to maximize the expected dollar value V from the treatment. Before making the treatment choice, there are n possible tests available, each with at most c possible results. Although we will evaluate the costs of the tests in dollars, the costs could also arise in practice from the delay in treatment or the side effects from performing the tests. At any point in time the doctor can make the treatment decision or order another test, knowing the results of all of the tests that have been previously ordered. Because the choice the doctor makes can depend on the results of those tests, the policy space of the test sequencing problem is exponential in the number of tests n, $(c+1)^n$, the number of possible sets of observations available in the decision making process.

Each test T_m has an associated cost $C_m > 0$, $m = 1, \ldots, n$, and we assume that the time horizon is short enough that its cost does not depend on when that particular test was performed. Therefore, the expected value of the prospect of any scenario is the difference between the dollar value of the treatment, $\mathrm{E}[V|T_x, T_1, \ldots, T_n]$, and the costs of the tests ordered, $C_1 + \cdots + C_n$. Such a value model is said to be *separable*, and we exploit that in our formulations. We assume, without loss of generality, that the same test will produce the same result if performed more than once, so it would never be optimal to order the same test twice. (A test that might be worth repeating could be included as multiple available tests.) The test sequencing problem is *general* as we impose no (conditional) independence assumptions on the n-vector of observable test results $\mathbf{R}$ conditioned on the disorder D, with probability distribution $\Pr\{\mathbf{R}|D\}$.

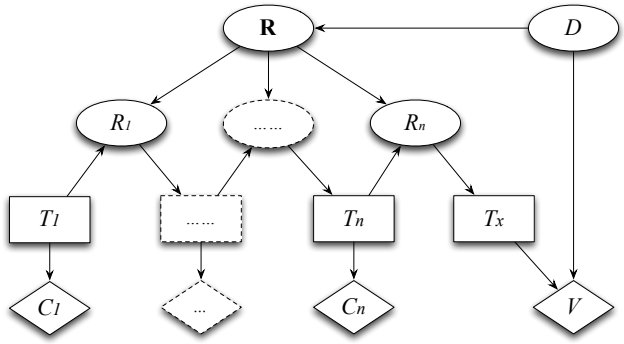

Figure 1: Influence diagram for the general test sequencing problem with n tests

An influence diagram for the general test sequencing problem is shown in Figure 1. There are n testing decisions, $T_1, \ldots, T_n$, each with $n+1$ alternatives (including not to test), and the treatment decision, T_x. Corresponding to each of the testing decisions T_m there is a dollar cost C_m of testing, depending on the test chosen, and an expected dollar value V that depends on both the disorder D and the treatment decision T_x, $\mathrm{E}[V|D, T_x]$. The potentially observable test results, $\mathbf{R}$, depends on D, and the actual test result R_m observed after testing decision T_m is therefore a deterministic function of T_m and $\mathbf{R}$,

$$R_m = \begin{cases} \mathbf{R}_{T_m} & \text{if } T_m \neq 0 \\ 1 & \text{if } T_m = 0 \end{cases} \quad \text{for } m = 1, \ldots, n. \qquad (1)$$

Earlier observations and decisions are known at the time of later decisions (Howard, 1977). Therefore, at the time of decision T_m the doctor will know which tests were ordered, $T_1, \ldots T_{m-1}$, and their corresponding results, $R_1, \ldots, R_{m-1}$. At the time of the treatment decision T_x the doctor will know all tests that were ordered, $T_1, \ldots, T_n$, and their results, $R_1, \ldots, R_n$. These definitions are summarized in Table 1. In the following sections we will consider different approaches to the exact solution of this problem.

3 INFLUENCE DIAGRAM SOLUTIONS

In this section we formulate an influence diagram solution to the general test sequencing problem and a more efficient formulation based on a Markov Decision Process (MDP) model. This method is symmetric in the sense that the probability distributions are full arrays and all variables are included in the formulations of every scenario.

Symbol	Definition
T_x	treatment decision
a	number of treatment alternatives
D	uncertain disorder
b	number of disorder states
V	expected dollar value of the treatment T_x for the disorder D
n	number of tests available
$\mathbf{R}$	n-vector of potentially observable test results
c	maximum number of possible results for each test
T_m	decision which test to order mth, $m = 1, \ldots, n$
R_m	results of the mth test ordered
C_m	cost of the mth test ordered
$\mathbf{S}_m$	n-vector of test results observed after m decisions

Table 1: **Symbol Definitions**

We can solve the influence diagram shown in Figure 1 by constructing a rooted cluster tree, as shown in Figure 2 (Shachter and Peot, 1992), similar for our purposes to a strong junction tree (Jensen et al., 1994), and minimal because these cliques are necessary to represent the problem (Shachter, 1999). The computational time complexity of the solution is determined by the total of the sizes of cluster tables, $abc^n + ac^{2n}(n+1)^n$, recognizing that the number of possible states for T_x, D, $\mathbf{R}$, T_m, and R_m are $a, b, c^n, n+1$, and c, respectively.

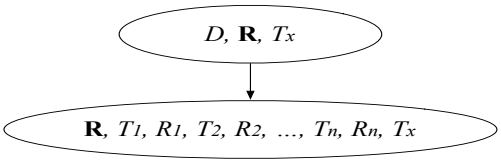

Figure 2: Rooted cluster tree for the influence diagram in Figure 1

Theorem 1. *The computational complexity of the standard influence diagram formulation of the general test sequencing problem is* $O(c^{2n}(n+1)^n)$.

An influence diagram formulation based on an MDP model, however, is more efficient than the standard influence diagram model for large n. The key is to introduce a *Markov state variable*, $\mathbf{S}_m$, the observed test results after m testing decisions, which renders past observations and decisions independent of future decisions. $\mathbf{S}_m$ is an n-vector, with components corresponding to the different possible test results, but with $c+1$ possible values for each component, including a new state "0" corresponding to the "test results not yet observed". Therefore, letting $\mathbf{S}_0 = \mathbf{0}$ indicate that no tests have been performed before the first testing decision, we can define $\mathbf{S}_m$ for each possible $j, m = 1, \ldots, n$ as a deterministic function of $\mathbf{S}_{m-1}$, T_m, and $\mathbf{R}$ by

$$(\mathbf{S}_m)_j = \begin{cases} \mathbf{R}_{T_m} & \text{if } T_m = j \neq 0 \\ (\mathbf{S}_{m-1})_j & \text{otherwise} \end{cases} . \qquad (2)$$

With this definition of $\mathbf{S}_m$ we can formulate the influence diagram shown in Figure 3. Because of the Markov state, the decisions $T_2, \ldots, T_n$ and T_x depend only on the corresponding state variables $\mathbf{S}_1, \ldots, \mathbf{S}_n$, respectively, rather than any of the past decisions and observations. However, this is not yet an MDP and would not be efficient to solve because of the role played by the uncertain potentially observable test results $\mathbf{R}$. Nevertheless, the definition of $\mathbf{S}_m$ allows us to reformulate this influence diagram into an MDP influence diagram without $\mathbf{R}$, as shown in Figure 4, that is efficient to solve.

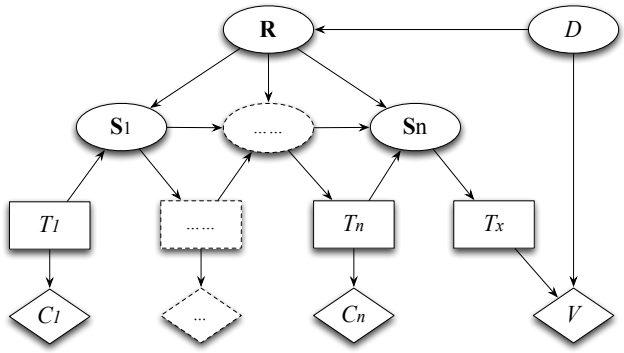

Figure 3: The influence diagram of the general test sequencing problem with Markov states

Theorem 2. *The influence diagram shown in Figure 4 is a valid representation of the general test sequencing problem.*

Proof. Given the relationships represented by the influence diagram shown in Figure 3, D is conditionally independent of $T_1, \ldots, T_n, \mathbf{S}_1, \ldots, \mathbf{S}_n$ given $\mathbf{R}$. By the definition of $\mathbf{S}_m$, $\mathbf{R}$ is conditionally independent of $T_1, \ldots, T_n, \mathbf{S}_1, \ldots, \mathbf{S}_{n-1}$ given $\mathbf{S}_n$. Therefore, it follows that D must be conditionally independent of $T_1, \ldots, T_n, \mathbf{S}_1, \ldots, \mathbf{S}_{n-1}$ given $\mathbf{S}_n$. Likewise, by the definition of $\mathbf{S}_m$, $\mathbf{S}_{m+1}$ is conditionally independent of $T_1, \ldots, T_m, \mathbf{S}_1, \ldots, \mathbf{S}_{m-1}$ given $\mathbf{S}_m$ and T_{m+1} for $m = 1, \ldots, n-1$, as shown in Figure 4. □

Even though the earlier testing decisions and their results will be known at the time of later decisions, it is sufficient to observe the Markov state $\mathbf{S}_m$ as shown in the MDP

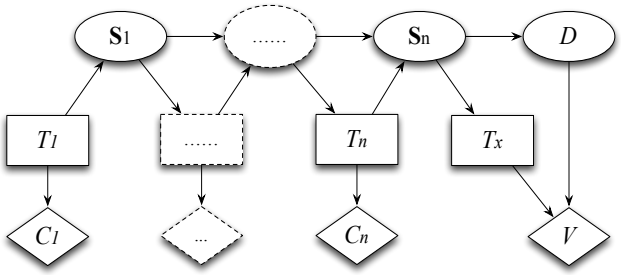

Figure 4: The MDP influence diagram for the general test sequencing problem

influence diagram (Figure 4). Therefore, we can solve it using the rooted cluster tree shown in Figure 5. As before, the computational time complexity of the solution is determined by the total of the sizes of cluster tables, $(n+1)(c+1)^n + (n-1)(n+1)(c+1)^{2n} + ab(c+1)^n$, recognizing that the number of possible states for T_x, D, T_m, and $\mathbf{S}_m$ are $a, b, n+1$, and $(c+1)^n$, respectively. We must also account for the cost of preprocessing to reformulate the diagram to the MDP at a complexity of $bc^n(c+1)^n + (n+1)(c+1)^{2n}$. Although this is dominated by the expression above, we include it in our final comparisons.

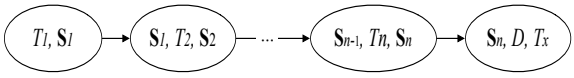

Figure 5: The rooted cluster tree for the MDP influence diagram

Theorem 3. *The computational complexity of the MDP influence diagram formulation of the general test sequencing problem is* $O(n^2(c+1)^{2n})$.

4 PURE DECISION TREE SOLUTION

In this section we formulate and analyze the computational complexity of the exact solution to the general test sequencing problem using an asymmetric pure decision analysis decision tree without *coalescence*, the reuse of sub-tree calculations. Such decision trees maintain the strict tree structure in which each node has at most one parent. Despite the prevalence and usefulness of decision tree models, there has been limited evaluation of their complexity when applied to asymmetric decision problems.

A decision tree is a natural representation for the asymmetry in the general test sequencing problem, recognizing that after we have observed m test results, there are only $n-m$ remaining tests to consider and the choice of which test to order, if any, can depend on the test results that we have already observed. To build a decision tree we will need to preprocess the probability distributions for D, $\mathbf{R}$, and $R_1, \ldots, R_n$ from the assessed distributions as shown in the influence diagram in Figure 1 to the inferential order they need to appear in the decision tree, where D and $\mathbf{R}$ are not observed before any of the decisions. The computational effort to perform this pre-processing is substantial, $O(bc(c+1)^n)$, but dominated by the work needed to evaluate the decision tree. Hence, we can ignore it in our analysis but include it in our final comparisons.

For each possible sequence of m tests and their observed test results, $m = 0, \ldots, n$, there is a decision node in the decision tree corresponding to the choice of treatment or of another test. There are $\binom{n}{m} m! c^m$ such possible sequences in the tree.

Proposition 1. *The total number of decision nodes in a pure decision tree without coalescence is*

$$\sum_{m=0}^{n} \binom{n}{m} m! c^m.$$

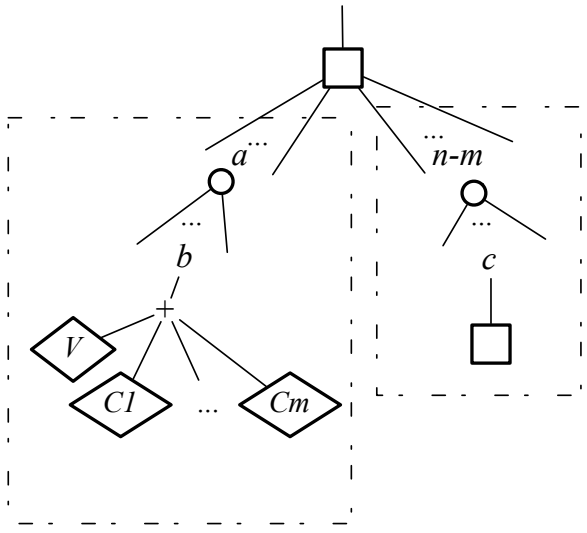

Figure 6: A generic decision node in the pure decision tree without coalescence, with m observed test results

The computational time complexity of the decision tree solution is determined by the number of arcs (or nodes) in the tree. Figure 6 shows a generic decision node within the decision tree whose ancestors include exactly m tests and their corresponding test results. There are $a+n-m$ alternatives, corresponding to choosing from one of the remaining $n-m$ tests or choosing to stop testing and make the treatment decision. For each of the test alternatives there are c possible test results, each leading to a different decision node in the tree, and for each of the a treatment alternatives there are b possible disorder states. Therefore, for each of the decision nodes in the tree there are

$a + ab + (n-m)c$ arcs in the tree, and the total number of arcs is given by

$$\begin{aligned}
& \sum_{m=0}^{n} \binom{n}{m} m! c^m [(a+ab) + (n-m)c] \\
= \;& (a+ab) \sum_{m=0}^{n} \frac{n!}{(n-m)!} c^m + \sum_{m=0}^{n} \frac{n!}{(n-m-1)!} c^{m+1} \\
= \;& (a+ab) c^n n! \sum_{m=0}^{n} \frac{c^{-m}}{m!} + c^n n! \sum_{m=0}^{n-1} \frac{c^{-m}}{m!} \\
\approx \;& (a+ab) c^n n! e^{1/c} + c^n n! e^{1/c} \\
= \;& (a+ab+1) c^n n! e^{1/c} = O(c^n n!)
\end{aligned}$$

Theorem 4. *The computational complexity of the decision tree formulation with no coalescence of the general test sequencing problem is* $O(c^n n!)$.

We will see in the next section that we can improve on the efficiency of the influence diagram and decision tree by allowing recursive coalescence, reusing subtree calculations as much as possible. Coalescence has traditionally been applied with decision trees in a limited fashion, at most once for any path in the tree (Howard, 1977; Olmsted, 1983), but we will apply it much more extensively in a generalization of decision trees called decision circuits.

5 DECISION CIRCUIT SOLUTION

In this section we formulate a decision circuit solution to the general test sequencing problem and show that it achieves the lower bound complexity of any algorithm for the problem that examines the entire policy space. Although this solution can be viewed as an extension of the pure decision tree solution with recursive coalescence, the decision circuit naturally integrates such coalescence and, unlike decision trees, does not need the distributions to be preprocessed.

5.1 INTRODUCTION TO DECISION CIRCUITS

Decision circuits are generalized decision trees that maintain their asymmetry while exploiting any conditional independence. They were developed by Bhattacharjya and Shachter (2007) and Bhattacharjya (2009) to efficiently evaluate influence diagrams with methods similar to arithmetic circuits for Bayesian networks (Darwiche, 2003). A decision problem represented by an influence diagram, or an intermediate structure, a decision circuit backbone, can be transformed into a decision circuit for efficient evaluation and sensitivity analysis (Shachter and Bhattacharjya, 2010; Bhattacharjya and Shachter, 2008, 2010). Although decision circuits were not developed as a representation for communication, Bhattacharjya and Shachter (2012) showed how formulating asymmetric decision circuits directly, instead of formulating an influence diagram and transforming it into a decision circuit, could be desirable in many applications. They also showed how to build decision circuits in assessed form, avoiding the probability distribution preprocessing effort (Bayes Theorem "tree flipping") needed for decision trees.

Decision circuits generalize decision trees in several key ways. Both decision circuits and decision trees are natural representations for asymmetric problems. Extensive, even recursive, coalescence is encouraged in the decision circuit, by allowing nodes to have multiple parents. Because the expectation operation for an uncertain variable in decision trees is represented as separate *sum* and *product* operations in the decision circuit, the probability distribution corresponding to the variable can appear further downstream (Shenoy, 1998). As a result, probability distributions can be incorporated into the decision circuit as assessed and there is no need for preprocessing the assessed distributions. Decision circuits are also able to exploit separable problem structure as found in influence diagrams and junction trees (Tatman and Shachter, 1990; Shachter and Peot, 1992; Jensen et al., 1994). Finally, once the probability and value distributions are specified the decision circuit can be compiled for even greater efficiency.

5.2 DECISION CIRUIT FORMULATION

An example of our decision circuit formulation is shown in Figure 7 for the general test sequencing problem with $a = b = c = n = 2$, that is, there are two treatment alternatives, two disorder states, and two tests available with two test results each. Decision circuits can have "indicator variables" λ to control and manage evidence and sensitivity analysis, but we have omitted them to simplify the diagram. Including the indicators would not significantly affect the computational complexity for the test sequencing problem.

At the leaves of the decision circuit shown in Figure 7 are the expected dollar values for each prospect $\mathrm{E}[V|D, T_x]$, the costs C for each type of test, the probabilities $\Pr\{D\}$, and the likelihoods $\Pr\{\mathbf{R}|D\}$. The separable costs are incorporated using *branching sum* ("B+") nodes (Shachter and Bhattacharjya, 2010). The quantities at the leaves are combined and reused throughout the circuit both as probabilities and as unnormalized probability-value hybrids for decision making. For example, we can marginalize for the case where some of the tests are not performed. The decision circuit is evaluated by sweeping up from the leaves to the root and, in the process, making all of the decisions at *max* nodes and determining the optimal expected value (Darwiche, 2003; Bhattacharjya and Shachter, 2007). A sweep down through the circuit computes derivatives of the optimal value with respect to any

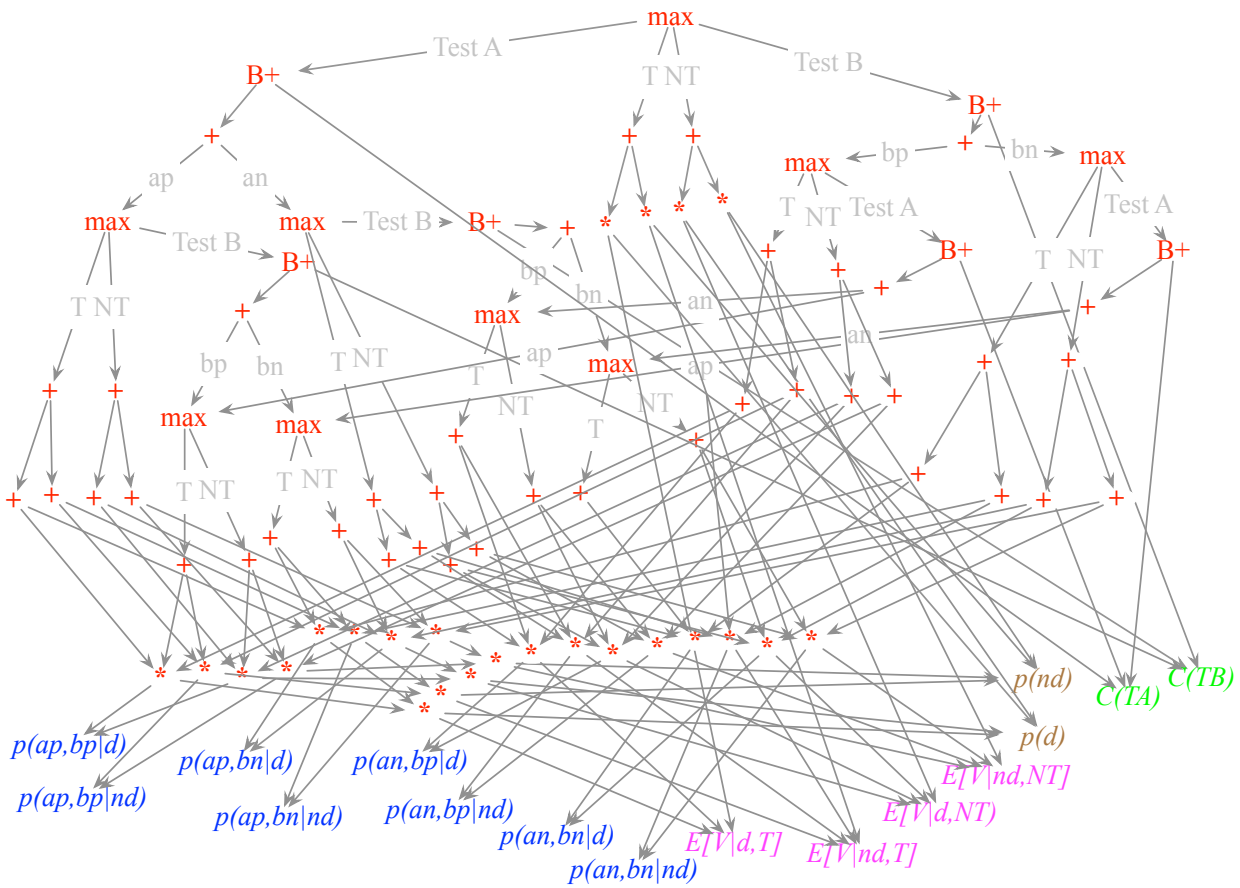

Figure 7: The decision circuit for general test sequencing involving two treatment alternatives, two disorder states, and two tests with two test results each

of the nodes and assessed parameters for use in sensitivity analysis (Bhattacharjya and Shachter, 2008, 2010). Therefore, we can compute the computational time complexity of a decision circuit by counting the number of arcs.

The coalescence in the decision circuit allows us to exploit essential properties of the general test sequencing problem. For example, when the results from m tests have been observed, the order those tests were performed does not matter (Jaynes, 2003; Bickel and Smith, 2006). In our decision circuit those $m!$ different test sequences (corresponding to m equivalent observed test results) all lead to the same decision (*max*) node for the selection of an $(m+1)$st test or to make the treatment decision. The structure of our circuit relies on the fact that the same choice will be optimal regardless what order the observed tests were performed, similar to the Markov state in our MDP formulation or an "information set" in game theory (Shenoy, 1998).

5.3 THE COMPLEXITY OF THE DECISION CIRCUIT FORMULATION

In our decision circuit formulation, for each possible set of m tests and their observed test results, $m = 0, \ldots, n$, there is a decision node in the decision tree corresponding to the choice of treatment or another test. There are $\binom{n}{m}c^m$ such possible sets in the circuit. Note that, by contrast, in the decision tree there was a distinction about the order of the tests, increasing the number of decision nodes by a factor of $m!$. The binomial theorem provides a closed-form expression for the number of decision nodes in the decision circuit.

Proposition 2. *The total number of decision nodes in the decision circuit is*

$$\sum_{m=0}^{n} \binom{n}{m} c^m = (c+1)^n.$$

The number of arcs in the decision circuit determines the computational time complexity for the circuit. Figure 8 shows a generic decision node within the decision circuit whose ancestors include exactly m tests and their cor-

responding test results. There are m arcs into the node, corresponding to the m different tests that could have been observed last. There are $a+n-m$ alternatives, corresponding to choosing from one of the remaining $n-m$ tests or choosing to stop testing and make the treatment decision. For each of the a treatment alternatives and b possible disorder states we marginalize over c possible test results (except when m is either 0 or n) to compute the probabilities and probability-value hybrids of the disease and observed test results from those used by decisions with $m+1$ test results. For each of the $n-m$ test alternatives there are three arcs into or out of the branching sum node used to register the separable cost of the test, followed by c test results, each leading to a different decision node in the circuit, and counted as incoming arcs for those nodes. Therefore, for each of the decision nodes in the circuit there are less than $m+(a+ab+abc)+3(n-m)$ arcs in the decision circuit. We will compute the complexity from each of these three terms separately.

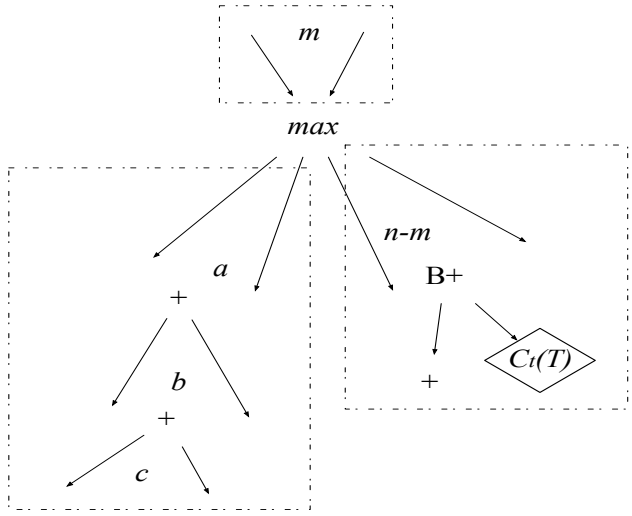

Figure 8: A generic decision node in the decision circuit with m observed test results

The first term corresponds to the m arcs directed into the decision node. The total for all decision nodes is

$$\begin{aligned}\sum_{m=0}^{n}\binom{n}{m}mc^m &= \sum_{m=1}^{n}\binom{n}{m}mc^m \\ &= \sum_{m=1}^{n}\frac{n!}{(m-1)!(n-m)!}c^m \\ &= nc\sum_{m=1}^{n}\frac{(n-1)!}{(m-1)!(n-m)!}c^{m-1} \\ &= nc(c+1)^{n-1}.\end{aligned} \tag{3}$$

The second term is the less than abc arcs corresponding to the treatment alternatives. The total for all decision nodes is less than

$$(a+ab+abc)\sum_{m=0}^{n}\binom{n}{m}c^m = (a+ab+abc)(c+1)^n. \tag{4}$$

The third term is the $3(n-m)$ arcs corresponding to the next tests and the arcs needed to register the cost of the test. The total for all decision nodes is

$$\begin{aligned}3\sum_{m=0}^{n-1}\binom{n}{m}(n-m)c^m &= 3\sum_{m=0}^{n-1}\frac{n!}{m!(n-m)!}(n-m)c^m \\ &= 3n\sum_{m=0}^{n-1}\frac{(n-1)!}{m!(n-1-m)!}c^m \\ &= 3n(c+1)^{n-1}.\end{aligned} \tag{5}$$

Finally, including the $2ab(c^n+1)$ arcs at the bottom of the decision circuit that incorporate the $\mathrm{E}[V|D,T_x]$, $\Pr\{D\}$, and $\Pr\{\mathbf{R}|D\}$ tables, the total number of arcs is less than

$$2ab(c^n+1)+n(c+3)(c+1)^{n-1}+(a+ab+abc)(c+1)^n,$$

and we have shown the following result.

Theorem 5. *The computational complexity of the decision circuit formulation of the general test sequencing problem is $O(n(c+1)^{n-1})$.*

In general the test sequencing problem can have an arbitrary optimal policy for any of the $(c+1)^n$ possible sets of observations. Given m tests have been performed and any of the c^m possible test results, the corresponding policy is determined by comparing the net expected values among the remaining $n-m$ tests that could be performed. The number of such comparisons is $\sum_{m=0}^{n-1}\binom{n}{m}(n-m)c^m$, similar to Equation 5, and this provides a lower bound on the complexity of any algorithm for the general problem that examines the entire policy space, $O(n(c+1)^{n-1})$. Because the decision circuit formulation achieves this bound, the bound must be tight.

Theorem 6. *Any algorithm for the general test sequencing problem that examines the entire policy space of test results must have time complexity with lower bound $\Omega(n(c+1)^{n-1})$, as achieved by the decision circuit formulation.*

We compare the computational complexity of the four models with the parameters $a=b=c$ set to 2 and 3. The results as a function of the number of tests available, n, are displayed in Table 2, and the logs of the complexities are shown in Figure 9 for $a=b=c=2$. The plots would appear similar, but with greater slopes, if we increased the values of the parameters a, b, and/or c.

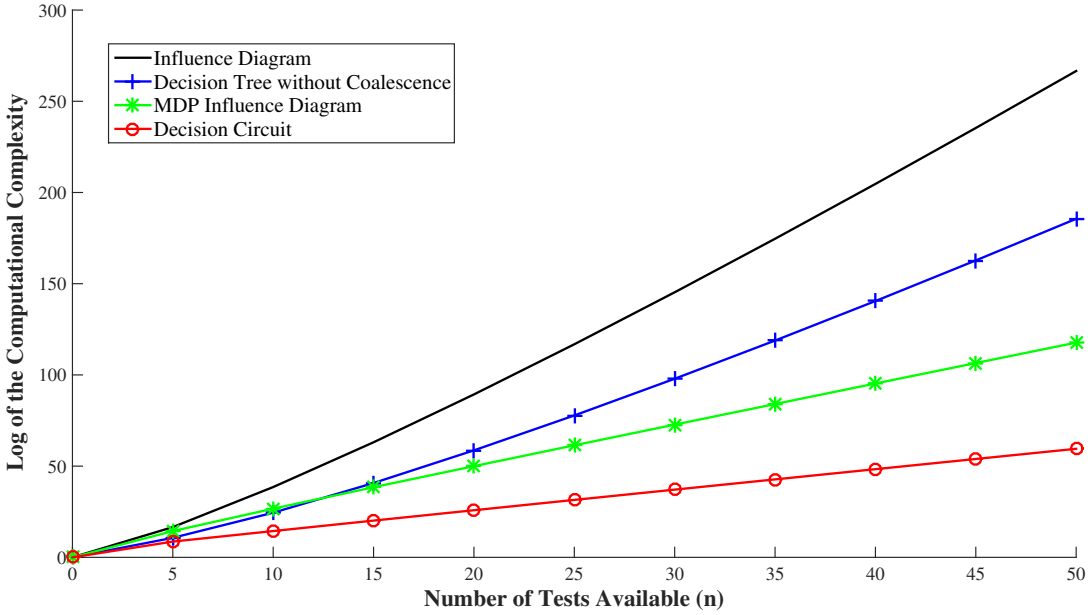

Figure 9: Log of the computational complexity of different formulations with model parameters $a = b = c = 2$

State Space	Formulation	5 Tests	10 Tests	15 Tests	20 Tests	25 Tests
$a = b = c = 2$	Decision Tree without Coalescence	4.43×10^{4}	4.29×10^{10}	4.95×10^{17}	2.94×10^{25}	6.00×10^{33}
	Standard Influence Diagram	1.59×10^{7}	5.44×10^{16}	2.48×10^{27}	6.12×10^{38}	5.33×10^{50}
	MDP Influence Diagram	1.79×10^{6}	3.84×10^{11}	4.94×10^{16}	5.11×10^{21}	4.67×10^{26}
	Decision Circuit	5.69×10^{3}	1.82×10^{6}	5.60×10^{8}	1.65×10^{11}	4.72×10^{13}
$a = b = c = 3$	Decision Tree without Coalescence	5.29×10^{5}	3.89×10^{12}	3.40×10^{20}	1.54×10^{29}	2.38×10^{38}
	Standard Influence Diagram	1.38×10^{9}	2.71×10^{20}	7.12×10^{32}	1.01×10^{46}	5.10×10^{59}
	MDP Influence Diagram	3.22×10^{7}	1.21×10^{14}	2.77×10^{20}	5.08×10^{26}	8.24×10^{32}
	Decision Circuit	5.20×10^{4}	5.77×10^{7}	6.63×10^{10}	7.59×10^{13}	8.61×10^{16}

Table 2: Computational complexity of different formulations with model parameters $a = b = c$

6 CONCLUSIONS

The general test sequencing problem has been known to be NP-hard with respect to the number of tests available, so most past research efforts have searched for approximate solution methods, heuristics, simulations, and simplifying assumptions.

In this paper, we develop a decision circuit formulation for the general test sequencing problem that solves the problem exactly without imposing any additional assumptions. We analyze its computational complexity and compare it with other frequently used sequential decision making models, pure decision trees and influence diagrams, and we develop a more efficient influence diagram model based on an MDP. In our comparison, the decision circuit model significantly outperforms the others. In fact, it achieves the lower bound on the computational complexity for any method for the general test sequencing problem that examines the entire policy space.

We could have obtained this same order of computational complexity by implementing decision trees with recursive coalescence. To construct such decision trees would have required explicit preprocessing of the assessed distributions, which is more efficiently done implicitly within our decision circuit formulation. It would also have used coalescence in a recursive manner natural in decision circuits but not common for decision trees.

In a similar fashion, we could have obtained a more efficient MDP formulation by recognizing the structural asymmetry in the problem. Because we observe at most one test result in each time period we could prune from

the state space those paths with more observations. This would improve the run-time computational complexity of the MDP and lead to a problem structure quite similar to the decision circuit.

Another advantage of using decision circuits is the efficient sensitivity analysis available on the assessed problem parameters, the probabilities and costs. The decision circuit formulation developed here allows us to solve the general test sequencing problem exactly for much larger values of n than has been possible before.

Information gathering decisions are strategic components in many decision problems in medicine, engineering, and other domains where they can significantly improve performance (Bickel and Smith, 2006). This paper has focused on the general test sequencing problem, with no assumptions, such as probabilistic independence. We realize that this is a specific class of problems, but we have used its structure to analyze and compare different exact techniques, and to demonstrate that a relatively new approach, decision circuits, allows us to achieve the complexity lower bound. This suggests that well-crafted decision circuits might perform relatively well on more general problems, such as troubleshooting, where observations and actions are interspersed and actions affect both the state of the system and future observations (Breese and Heckerman, 1996).

Acknowledgements

We thank the anonymous referees for their suggestions and Yuval Shahar for inspiring the research.

References

Richard Bellman. A problem in the sequential design of experiments. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 16(3/4):221–229, 1956.

Debarun Bhattacharjya. *Decision circuits: A graphical representation for efficient decision analysis computation*. PhD thesis, Stanford University, 2009.

Debarun Bhattacharjya and Ross D Shachter. Evaluating influence diagrams with decision circuits. In *Proceedings of the Twenty-Third Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 9–16, 2007.

Debarun Bhattacharjya and Ross D Shachter. Sensitivity analysis in decision circuits. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 34–42, 2008.

Debarun Bhattacharjya and Ross D Shachter. Three new sensitivity analysis methods for influence diagrams. In *Proceedings of the Twenty-Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 56–64, 2010.

Debarun Bhattacharjya and Ross D Shachter. Formulating Asymmetric Decision Problems as Decision Circuits. *Decision Analysis*, 9(2):138–145, 2012.

Eric J Bickel and James E Smith. Optimal sequential exploration: A binary learning model. *Decision Analysis*, 3(1):16–32, 2006.

Concha Bielza and Prakash P Shenoy. A Comparison of Graphical Techniques for Asymmetric Decision Problems. *Management Science*, 45(11):1552–1569, 1999.

Concha Bielza, Manuel Gómez, and Prakash P Shenoy. A review of representation issues and modeling challenges with influence diagrams. *Omega*, 39(3):227–241, 2011.

John S Breese and David Heckerman. Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, 1996.

Zvi Covaliu and Robert M. Oliver. Representation and solution of decision problems using sequential decision diagrams. *Management Science*, 41(12):pp. 1860–1881, 1995.

Adnan Darwiche. A differential approach to inference in Bayesian networks. *J. ACM*, 50(3):280–305, 2003.

Riza Demirer and Prakash P Shenoy. Sequential valuation networks for asymmetric decision problems. *European Journal of Operational Research*, 169(1):286–309, 2006.

Ronald A Howard. The used car buyer. In *Readings in Decision Analysis*. Stanford Research Institute, Menlo Park, CA, 1977.

Ronald A Howard and James E Matheson. Influence Diagrams. *R. A. Howard, J. E. Matheson, eds. The Principles and Applications of Decision Analysis. 1984*, 2: 719–762, 1981.

Edwin T Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

Finn V Jensen and Marta Vomlelová. Unconstrained influence diagrams. In *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 234–241, 2002.

Frank Jensen, Finn V Jensen, and Søren L Dittmer. From influence diagrams to junction trees. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 367–373. Morgan Kaufmann Publishers Inc., 1994.

Scott M Olmsted. *On Representing and Solving Decision Problems*. PhD thesis, Stanford University, 1983.

Christos H Papadimitriou and John N Tsitsiklis. The Complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12:441–450, 1987.

Ross D Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

Ross D. Shachter. Efficient value of information computation. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 594–601, 1999.

Ross D Shachter and Debarun Bhattacharjya. Dynamic programming in influence diagrams with decision circuits. In *Proceedings of the Twenty-Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 509–516, 2010.

Ross D. Shachter and Mark A. Peot. Decision making using probabilistic inference methods. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI-92)*, pages 276–283, 1992.

Prakash P Shenoy. Game Trees For Decision Analysis. *Theory and decision*, 44(2):149–171, 1998.

Prakash P Shenoy. Valuation network representation and solution of asymmetric decision problems. *European Journal of Operational Research*, 121(3):579–608, 2000.

James E Smith, Samuel Holtzman, and James E Matheson. Structuring conditional relationships in influence diagrams. *Operations Research*, 41(2):280–297, 1993.

Joseph A Tatman and Ross D Shachter. Dynamic programming and influence diagrams. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):365–379, 1990.

Canan Ulu and James E Smith. Uncertainty, information acquisition, and technology adoption. *Operations Research*, 57(3):740–752, 2009.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.