

Cost-effectiveness analysis with influence diagrams

M. Arias and F. J. Díez
Dept. Artificial Intelligence, UNED, Madrid, Spain.
Phone: +34-91-3988743/3987161.
Fax: +34-91-3989382
E-mail: marias/fjdiez@dia.uned.es.

June 2, 2014

Abstract

Introduction: Cost-effectiveness analysis (CEA) is used increasingly in medicine to determine whether the health benefit of an intervention outweighs the economic cost. Decision trees, the standard decision modeling technique for non-temporal domains, can only perform CEAs for very small problems.

Objective: To develop a method for CEA in problems involving several dozen variables.

Methods: We explain how to build influence diagrams (IDs) that explicitly represent cost and effectiveness. We propose an algorithm for evaluating cost-effectiveness IDs directly, i.e., without expanding an equivalent decision tree.

Results: The evaluation of an ID returns a set of intervals for the willingness to pay—separated by incremental cost-effectiveness ratios (ICERs)—and, for each interval, the cost, the effectiveness, and the optimal intervention. The algorithm that evaluates the ID directly is in general much more efficient than the brute-force method, which is in turn more efficient than the expansion of an equivalent decision tree. Using OpenMarkov, an open-source software tool that implements this algorithm, we have been able to perform CEAs on two medical IDs whose equivalent decision trees contain billions of branches.

Conclusion: IDs can perform CEA on many problems that could not be analyzed with decision trees.

Keywords: cost-benefit analysis, cost effectiveness, decision trees, influence diagrams.

1 Introduction

In medicine, many new interventions become available every year for the diagnosis, prevention, and treatment of different diseases.¹ Given that economic resources are limited, it is necessary to assess whether the health benefit of an intervention is worth its cost. The more common approach to economic evaluation is cost-effectiveness analysis (CEA) [1, 2]. In this context, the *net monetary benefit* [3] of an intervention I is

$$NMB_I(\lambda) = \lambda \cdot e - c, \quad (1)$$

where e is its effectiveness and c its cost. The parameter λ , usually called *willingness to pay*, *cost-effectiveness threshold* or *ceiling ratio*, converts effectiveness into a monetary scale. It takes values on the set of positive real numbers, i.e., on the interval $(0, +\infty)$. It is measured in effectiveness units divided by cost units; for example, in dollars per death avoided or euros per quality-adjusted life year (QALY) [4]. As the willingness to pay is different for each decision maker, CEA must consider all the possible values. The result of the analysis is usually a set of intervals for λ , each one having an optimal intervention.

When the consequences of the interventions are not deterministic, it is necessary to build a model that takes into account the probability of each outcome. Decision trees are the tool used most frequently for modeling decision problems with uncertainty, especially in medicine [5]. Their main drawback is that their size grows exponentially with the number of variables.² In the medical literature, trees usually have 3 or 4 variables and between 6 and 10 leaf nodes. A tree of 5 variables typically contains around 20 leaf nodes, which implies that building, debugging, and analyzing it would require a significant effort. Given that the maximum size of decision trees in practice is of the order of 50 nodes, they can only solve problems of at most 6 or 7 variables.

Influence diagrams (IDs) [7, 5], in contrast, have the advantages of being very compact, representing conditional independencies, and using direct probabilities, i.e., the probability of the effect conditioned on the cause. They have been used to solve complex medical problems. For example, Arthronet [8], an ID for knee arthroplasty, contains 23 variables and Mediastinet [9, 10], for lung cancer, 27.³

Building an ID that represents cost and effectiveness is easy, but evaluating it is not. Standard methods, such as arc reversal [11, 12] and variable elimination [13], can only evaluate unicriterion IDs. An approximate solution would be to combine cost and effectiveness into a single criterion, the NMB (cf. Eq. 1), and evaluate the ID for several values of λ . The problem is that the more precision we require, the more values of λ we must examine; finding the exact solution would need infinite evaluations. Another approach would be to expand the ID into an equivalent decision tree [7] and evaluate it as proposed in [6]. The problem is that the size of the tree grows exponentially with the number of variables. When trying to convert the unicriterion versions of Arthronet and Mediastinet into decision trees, our computer ran out of memory, even though arc reversal and variable elimination were able to evaluate those IDs in a fraction of a second. The decision trees for the cost-effectiveness versions of those diagrams would run out of memory even faster, as we will discuss in Section 4.2.

In this paper we propose a new method for evaluating cost-effectiveness IDs, which combines arc reversal with the same idea used in [6] to evaluate cost-effectiveness decision trees: given that the values of λ are infinite and it is impossible to evaluate sequentially an infinite number of IDs, our algorithm groups the λ 's having the same cost, effectiveness, and optimal intervention into a finite set of intervals, whose boundaries are determined dynamically during the evaluation of the ID. This algorithm is in general much more efficient than the brute-force method, which is in turn more efficient than the expansion of an equivalent decision tree. After implementing the algorithm in OpenMarkov, an open-source software package for probabilistic graphical models, which is available at www.openmarkov.org, we have been able to evaluate several cost-effectiveness IDs, including Arthronet and Mediastinet, which contain dozens of variables.

The rest of this paper is structured as follows: Section 2 reviews the basic concepts of CEA and IDs. Section 3 presents the new algorithm, Section 4 compares IDs with decision trees from the point of view of knowledge representation and that of the computational complexity, Section 5 analyzes some related work, and Section 6 summarizes the conclusions. The appendix contains some formal definitions, the pseudo-code of the algorithms, and a detailed example.

¹By “intervention” we mean either a single action, such as applying a therapy, or a whole strategy, such as “Do the test T ; if it is positive, apply drug D every 8 hours; if it is negative, repeat the test after 24 hours and then...”.

²A variable represents, for instance, the presence of a disease, the result of a test, or a decision about a treatment. In general a variable gives rise to several nodes in the decision tree—see [6].

³These IDs, encoded in the ProbModelXML format, are publicly available at www.probmodelxml.org/networks.

2 Background

2.1 Deterministic CEA

When we have a set of interventions such that their cost and effectiveness are known with certainty, we can perform a deterministic CEA, which returns the intervention that maximizes the NMB (cf. Eq. 1) for each value of λ .

The result of a CEA can be expressed as a cost-effectiveness partition (CEP), which consists of a set of intervals such that the cost, effectiveness, and optimal intervention is the same for all the λ 's inside an interval, as shown in Figure 1.

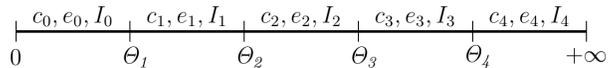


Figure 1: A cost-effectiveness partition (CEP) of four thresholds (five intervals). All the λ 's inside the i -th interval have the same cost (c_i), effectiveness (e_i), and optimal intervention (I_i).

The standard algorithm for deterministic CEA consists of finding a subset of interventions that dominates the other interventions for every value of λ [14]. A slightly more efficient algorithm is presented in [6] and in the appendix.

2.2 Decision trees

A decision tree is a model for decision analysis of problems involving probabilistic outcomes [15]. It has three types of nodes: chance, decision, and utility. Utility nodes represent the decision maker's preferences. All the leaves are utility nodes and, conversely, all utility nodes are leaves.

A unicriterion decision tree can be evaluated with the roll-back algorithm, which proceeds from the leaves to the root: the utility of a chance node is the average of the utilities of its branches and the utility of a decision node is the maximum of their utilities.

It is possible to build a bi-criteria decision tree such that each utility node, instead of representing a single value, represents cost and effectiveness separately. If the only decision node is the root, the tree can be evaluated by a modified version of the roll-back algorithm that computes the cost and effectiveness of each node separately and then performs a deterministic CEA at the root node [1, 16].

A tree containing embedded decision nodes can be evaluated with a modified version of the roll-back algorithm that works with CEPs instead of scalar utilities: each chance node is evaluated by averaging the CEPs of its branches, and each decision node is evaluated by performing a deterministic CEA for each interval of λ [6]. This method is the basis for the evaluation of IDs proposed in this paper.

2.3 Unicriterion influence diagrams

An ID is a probabilistic graphical model that consists of an acyclic directed graph containing three types of nodes—chance, decision, and utility—, a probability distribution and one or several utility functions. Every node represents a variable, so we will speak of nodes and variables indifferently. In the rest of the paper, an uppercase letter (X) denotes a variable or a node, and a lowercase letter (x) a value of the corresponding variable. A bold uppercase letter denotes a set of variables, $\mathbf{X} = \{X_1, \dots, X_n\}$, and a bold lowercase letter a configuration of the corresponding set of variables: $\mathbf{x} = (x_1, \dots, x_n)$.

In a graph, if there is an arc from node X to node Y , we say that X is a parent of Y ; the set of parents of Y is denoted by $Pa(Y)$, and $pa(Y)$ denotes a configuration of $Pa(Y)$. For every chance node C the ID contains a family of conditional probability distributions of the form $P(c|pa(C))$. A unicriterion ID also contains a utility function $U_j(pa(U_j))$ for each utility node U_j .

The **meaning** of an arc in an ID depends on the type of nodes that it links. An arc $X \rightarrow C$, where C is a chance node, denotes a probabilistic dependence of C on X ; in practice, it usually means that X is a cause of C . An arc from a decision D_i to a decision D_j (in general, any directed path from D_i to D_j) means that D_i is made before D_j . Every ID must contain a directed path passing through all the decision nodes, which determines a total ordering of the decisions. An arc from a chance node C to a decision node D_j means that the value of variable C is known when making decision D_j . IDs assume

the *non-forgetting hypothesis*, which means that a variable C known for a decision is also known for any posterior decision. An arc from a variable X to a utility node U_j means that the utility depends on X .

The *maximum expected utility (MEU)* of an ID whose chance and decision variables are all discrete is

$$MEU = \sum_{\mathbf{c}_0} \max_{d_1} \sum_{\mathbf{c}_1} \dots \sum_{\mathbf{c}_{n-1}} \max_{d_n} \sum_{\mathbf{c}_n} \prod_{C \in \mathbf{V}_C} P(c|pa(C)) \cdot \sum_j U_j(pa(U_j)), \quad (2)$$

where \mathbf{C}_i denotes the set of variables known after making D_i and before making D_{i+1} . Due to the non-forgetting hypothesis, the variables known when making D_i are $\mathbf{C}_0 \cup \{D_1\} \cup \mathbf{C}_1 \cup \dots \cup \{D_{i-1}\} \cup \mathbf{C}_{i-1}$. The optimal policy for D_i consists of selecting, for each configuration of these variables, the value d_i that maximizes the expected utility.

Arc reversal [11, 12] is the standard algorithm for evaluating unicriterion IDs; in this context, arc and link are synonymous. Reversing an arc $X \rightarrow Y$ (where X and Y are chance variables) consists of replacing it with $Y \rightarrow X$, computing new conditional probabilities for X and Y , and adding some links if necessary, as described in the appendix. The algorithm proceeds by removing a node in each iteration. When no node can be removed immediately, there is always a sequence of arc reversals that make one node removable. Section 4.2 discusses the complexity of this and other algorithms for IDs.

3 Cost-effectiveness analysis with IDs

This section describes our method for performing CEAs with IDs.

3.1 Construction of the ID

The construction of an ID for CEA is almost identical to the unicriterion case; the only difference is that some utility nodes represent effectiveness and the others represent economic costs.

Example 1 *For a disease whose prevalence is 0.14, there are two possible therapies. The effectiveness of each therapy depends on whether the disease is present or not, as shown in Table 1. There is a test with a sensitivity of 90%, a specificity of 93%, and a cost of 150 €. Is the test cost-effective?*

Therapy	Cost	Effectiveness	
		+disease	-disease
no therapy	0 €	1.2 QALY	10.0 QALY
therapy 1	20 000 €	4.0 QALY	9.9 QALY
therapy 2	70 000 €	6.5 QALY	9.3 QALY

Table 1: Cost and effectiveness of each intervention for the Example 1.

The ID for this example is shown in Figure 2. Decisions are drawn as rectangles, chance nodes as rounded rectangles, and utility nodes as hexagons. The node *Dec:Test* represents the decision whether or not to do the test; it has two values: *do test* and *do not test*. The node *Test* represents the result of the test; its values are *positive*, *negative*, and *not performed*.

3.2 Evaluation of cost-effectiveness IDs

3.2.1 Initial step: CEP potential for the utility node

The first step of the evaluation consists of combining all the utility nodes into a single node, U , whose parents are all the chance and decision nodes having at least one utility node in the original ID; we denote this set by \mathbf{X} . The utility for node U is a CEP potential, i.e., a function $U(\mathbf{x})$ that assigns a CEP to each configuration \mathbf{x} . In this initial ID, every CEP $U(\mathbf{x})$ consists of a single interval, because no threshold has been generated so far; the cost is the sum of the costs in the original ID for configuration \mathbf{x} and the effectiveness is the sum of the effectiveness values.

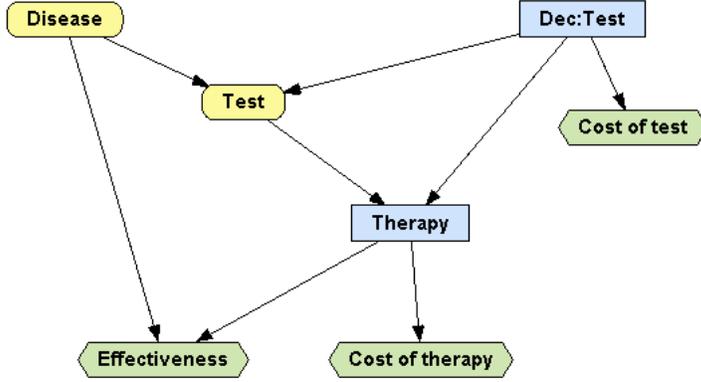


Figure 2: Cost-effectiveness influence diagram for the Example 1.

3.2.2 Arc reversal and node removal

After this initial step, the arc reversal algorithm proceeds by removing all the chance and decision nodes, as in the evaluation of unicriterion IDs [11, 12]; the only difference is that now utility tables are potentials of CEPs instead of potentials of real numbers.

A chance or a decision node having no children can be removed without more ado.

When removing a chance node C having U as the only descendant, the old parents of C become parents of U and the new CEP potential for U is computed as follows: for each configuration \mathbf{v} of the new parents of U we collect all the corresponding CEPs in the old potential and find the union of all their thresholds. In each interval we average the costs and then the effectiveness values, weighted by the conditional probabilities of C —a probability for each value c .

Similarly, when removing a decision node D the old parents of D become parents of U in the new ID. The new CEP potential for U is computed as follows: for each configuration \mathbf{v} of the new parents of U we collect all the corresponding CEPs in the old potential for U and collect all their thresholds. In each interval we perform a deterministic CEA, which may originate new thresholds. It is also possible that some of the old thresholds become unnecessary because the subintervals that they separate have the same cost, effectiveness, and optimal policy. Those thresholds should be removed as soon as they are detected.

When it is not possible to remove any node immediately, there is always a sequence of arc reversals that make at least one node removable. Reversing an arc in a cost-effectiveness ID is exactly the same as in the unicriterion case.

After removing all the chance and decision nodes, the potential of U contains only one CEP, which is the solution of the CEA.

The appendix contains the pseudo-code for this algorithm and a detailed numerical example.

4 Comparison with decision trees

4.1 Construction of the model: influence diagrams vs. decision trees

The main advantage of IDs is that they are much easier to build than decision trees, mainly because they are much more compact: an ID contains only one node for each variable in the problem, while in a tree the number of branches grows exponentially with the number of variables. For example, the ID in Figure 2 contains 6 nodes while the equivalent tree contains 32—see [6]. A symmetric decision tree containing n binary variables has 2^n leaf nodes. (A tree is said to be *symmetric* if every path from the root to a leaf node contains the same variables and in the same order, and all the nodes representing a variable have the same number of outgoing branches.) The number of leaves may be smaller due to asymmetries, but it may also be bigger if there are multi-valued variables. For this reason we said in the introduction that decision trees are unsuitable for problems involving more than 6 or 7 variables.

Compared with decision trees, ID do not only require much fewer parameters: these parameters are much easier to obtain because each conditional probability only involves one node (the effect) and its parents in the graph (its causes). These probabilities can be obtained from objective data, such as databases or epidemiological studies, or from subjective estimates. In contrast, the distribution probability for the branches of a node in a tree is usually conditioned on several of its causes and effects; when the tree contains more than a few variables, obtaining these conditional probabilities is extremely difficult if not impossible.

This explains why many medical problems for which it was unfeasible to build a decision tree directly have been solved with IDs; in the introduction we have mentioned two examples—Arthronet and Mediastinet—and there are many more [5].

4.2 Complexity of the evaluation algorithms

4.2.1 Influence diagrams vs. decision trees

The evaluation of unicriterion IDs, closely related to the evaluation of Bayesian networks, is NP-hard [17], which makes it unlikely that there exist polynomial complexity algorithms to evaluate them. Both the brute-force method, which consists of the direct application of Equation 2, and the transformation of the ID into an equivalent tree have exponential complexity in time and space. However, the latter is more inefficient than brute-force because it requires more space to store the structure of the tree and more time to compute the conditional probabilities of its branches.

Arc reversal and other exact algorithms for IDs, such as variable elimination [13] and strong junction trees [18], also have worst-case exponential complexity, both in time and space, but in many cases they are much more efficient. In fact, the property that most affects the complexity of these algorithms is not the number of nodes, but the treewidth of the graph (after removing the information arcs), a concept that has been extensively studied in Bayesian networks [19, 20, 21, 22]. Given that the treewidth of an ID is usually much smaller than the number of variables—as is also the case for Bayesian networks—these algorithms are in general much more efficient than the brute-force method and, consequently, much more efficient than the expansion of an equivalent tree.

In the case of cost-effectiveness analysis, the complexity of CEPs operations is proportional to the number of thresholds (see Algorithms 2 and 3 in the appendix), hence the difference is even more favorable for IDs, which perform fewer CEP operations than decision trees.

4.2.2 Evaluation of cost-effectiveness IDs

A deterministic CEA (cf. Sec. 2.1) involving n interventions may generate up to $n - 1$ thresholds. The combination of CEPs generates a new CEP that may inherit the thresholds of all of them. Therefore, the number of thresholds might grow exponentially with the number of variables. However, in practice this number depends on the numerical parameters (the probabilities and utilities) of the model. Thus, when a subset of k interventions dominates the others, only $k - 1$ thresholds are generated. In many cases k is smaller than the number of interventions; frequently $k = 1$, which generates no new threshold. Additionally, a significant number of thresholds are usually discarded when removing a decision variable (see line 7 of Algorithm 3 in the appendix). For these reasons, the number of thresholds often remains within manageable limits. For example, the evaluation of the ID in Figure 2 might generate up to 12 thresholds, but the maximum number during the evaluation is 3, and the final CEP contains only 2 (see the appendix).

As mentioned above, the complexity of CEPs operations is proportional to the number of thresholds. If the average of thresholds per CEP during the evaluation of an ID is k , then the computational cost the evaluation is roughly k times the cost of evaluating its unicriterion version. Thus, OpenMarkov is able to evaluate the unicriterion version of Mediastinet (with $\lambda = 30,000$ €/QALY, which is the reference value in Spain) in 18 ms on a desktop computer; the evaluation of the cost-effectiveness version needs 57 ms, only 3.13 times more. The evaluation of Arthronet requires 38 ms and 266 ms respectively, 6.93 times more.⁴ The evaluation of these IDs is very fast, even though the equivalent decision trees have billions of branches. Thus, if we had to perform a sensitivity analysis for any of these networks, we might run 10,000 simulations in less than an hour—or in some minutes in a parallel implementation. Given

⁴These four networks are available at www.probmodelxml.org/networks. The web page www.openmarkov.org/cea explains how to evaluate them.

that cost-effectiveness analyses are usually performed off-line (to inform health policies rather than to make bedside clinical decisions), the problems that can be solved with this method are much bigger than those that could be solved using decision trees, especially because it was generally accepted that it was not possible to perform CEAs with decision trees containing embedded decision nodes [23], until a new method was proposed recently [6].

5 Related work

To the best of our knowledge, our algorithm is the first method for solving cost-effectiveness IDs. Previously the only technique for analyzing multi-attribute IDs was the method by Nielsen et al. [24]: they select a particular configuration of α —the vector that represents the decision maker’s preferences—and evaluate the corresponding unicriterion ID; then they obtain a set of inequalities, which delimit the region of \mathbb{R}^n in which α can vary without leading to a different optimal strategy. This approach is clearly insufficient for cost-effectiveness analysis, especially in medicine, because it does not return the intervals for λ nor the optimal strategy for each interval. In contrast, the algorithm proposed in this paper limits itself to bicriteria IDs, but is able to solve them completely.

6 Conclusion and future work

In this paper we have explained how to build IDs for cost-effectiveness analysis, which can solve many medical problems for which it would be absolutely impossible to build a decision tree directly. In particular, our research group has built cost-effectiveness IDs for two complex medical problems [8, 9, 10]; the equivalent decision trees would contain billions of branches.

Additionally, we have adapted the arc reversal algorithm [11, 12] to evaluate cost-effectiveness IDs by combining it with a method for performing CEA in decision trees [6]. It is straightforward to adapt it to other algorithms for unicriterion trees, such as variable elimination [13] and strong junction trees [18], because all of them perform essentially the same operations. These algorithms are in general much more efficient in time and space than expanding the ID into an equivalent decision tree. In particular, after implementing this algorithm in OpenMarkov, an open source tool for probabilistic graphical models, we were able to perform CEAs on the IDs mentioned above, whereas expanding the equivalent decision tree was impossible even for their unicriterion versions because our computer ran out of memory. This implementation, together with the IDs mentioned in this paper, is publicly available at www.openmarkov.org/cea.

An open line for future research is to summarize into a small policy tree the result of evaluating a cost-effectiveness ID, by adapting the method proposed in [9]. Another line is to adapt our CEA algorithms to the evaluation of decision analysis networks (DANs) [25], which present several advantages over IDs, especially in the case of asymmetric decision problems.

Acknowledgments

This work has been supported by grants TIN2006-11152 and TIN2009-09158, from the Spanish Ministry of Science and Technology, by FONCICYT grant 85195, and by grant PI13/02446 of the Health Institute Carlos III. We would also like to thank the reviewers for their valuable comments and suggestions.

References

- [1] Drummond MF, Sculpher MJ, Torrance GW, O’Brien BJ, Stoddart GL. *Methods for the Economic Evaluation of Health Care Programmes*. 3rd ed. Oxford University Press; 2005.
- [2] Gold MR, Siegel JE, Russell LB, Weinstein MC. *Cost-Effectiveness in Health and Medicine*. New York: Oxford University Press; 1996.
- [3] Stinnett AA, Mullahy J. Net health benefit: A new framework for the analysis of uncertainty in cost-effectiveness analysis. *Medical Decision Making*. 1998;18:S68–S80.
- [4] Weinstein MC, Torrance G, McGuire A. QALYs: The basics. *Value in Health*. 2009;12:S5–S9.

- [5] Pauker S, Wong J. The influence of influence diagrams in medicine. *Decision Analysis*. 2005;2:238–244.
- [6] Arias M, Díez FJ. Cost-effectiveness analysis with sequential decisions. Madrid, Spain: UNED; 2011. Available from: www.cisiad.uned.es/techreports/cea-multidec.php.
- [7] Howard RA, Matheson JE. Influence diagrams. In: Howard RA, Matheson JE, editors. *Readings on the Principles and Applications of Decision Analysis*. Menlo Park, CA: Strategic Decisions Group; 1984. p. 719–762.
- [8] León D. A Probabilistic Graphical Model for Total Knee Arthroplasty. Master’s thesis. Dept. Artificial Intelligence, UNED. Madrid, Spain; 2011.
- [9] Luque M. Probabilistic Graphical Models for Decision Making in Medicine. PhD thesis. UNED. Madrid; 2009.
- [10] Luque M, Díez FJ, Disdier C. A decision support system for the mediastinal staging of non-small cell lung cancer. In: 31st Annual Meeting of the Society for Medical Decision Making. Los Angeles, CA; 2009. .
- [11] Olmsted SM. On Representing and Solving Decision Problems. PhD thesis. Dept. Engineering-Economic Systems. Stanford University, CA; 1983.
- [12] Shachter RD. Evaluating influence diagrams. *Operations Research*. 1986;34:871–882.
- [13] Jensen FV, Nielsen TD. *Bayesian Networks and Decision Graphs*. 2nd ed. New York: Springer-Verlag; 2007.
- [14] Weinstein MC, Stason WB. Foundations of cost-effectiveness analysis for health and medical practices. *New England Journal of Medicine*. 1977;296:716–721.
- [15] Weinstein MC, Fineberg HV, Elstein AS, et al. *Clinical Decision Analysis*. Philadelphia, PA: Saunders; 1980.
- [16] Sox HC, Blatt MA, Higgins MC, Marton KI. *Medical Decision Making*. Woburn, MA: Butterworth-Heinemann; 1988.
- [17] Cooper GF. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*. 1990;42:393–405.
- [18] Jensen F, Jensen FV, Dittmer SL. From influence diagrams to junction trees. In: de Mantaras RL, Poole D, editors. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI’94)*. San Francisco, CA: Morgan Kauffmann; 1994. p. 367–373.
- [19] Chavira M, Darwiche A. Compiling Bayesian networks using variable elimination. In: Veloso MM, editor. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI’07)*. Hyderabad, India: Morgan Kauffmann; 2007. p. 2443–2449.
- [20] Darwiche A. Recursive conditioning. *Artificial Intelligence*. 2001;126:5–41.
- [21] Koller D, Friedman N. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press; 2009.
- [22] Kwisthout J, Bodlaender H, van der Gaag L. The necessity of bounded treewidth for efficient inference in Bayesian networks. In: *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI’10)*. Lisbon, Portugal; 2010. p. 237–242.
- [23] Kuntz KM, Weinstein MC. Modelling in economic evaluation. In: Drummond MF, McGuire A, editors. *Economic Evaluation in Health Care*. New York: Oxford University Press; 2001. p. 141–171.
- [24] Nielsen SH, Nielsen TD, Jensen FV. Multi-currency influence diagrams. In: Salmerón A, Gámez JA, editors. *Advances in Probabilistic Graphical Models*. Berlin, Germany: Springer; 2007. p. 275–294.
- [25] Díez FJ, Luque M, König C. Decision analysis networks. In: Cano A, Gómez M, Nielsen TD, editors. *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM’12)*. Granada, Spain; 2012. p. 83–90.

A Appendix

This paper contains a brief description of the arc reversal algorithm, a formal definition of CEP, the pseudo-code for CEA algorithms, and a detailed solution of the problem stated in Example 1. It may be useful to compare it with the decision tree solution [6].

A.1 Arc reversal for unicriterion IDs

The arc reversal algorithm [11, 12] assumes that the ID has only one utility node, U ; otherwise, all the utility nodes must be summed up into a single node. The four basic operations of the algorithm are:

1. **Barren node removal.** A node is said to be *barren* if it has no children. Barren nodes can be removed from the ID without performing any additional operation.
2. **Chance node removal.** A chance node C whose only child is the utility node U can be removed from the ID by drawing links from each parent of C to U . If $P_{\text{old}}(c|pa(C))$ is the conditional probability of C and $U_{\text{old}}(pa(U))$ is the utility potential before eliminating C , the new utility potential after eliminating C is

$$U_{\text{new}}(\mathbf{v}) = \sum_c P_{\text{old}}\left(c|\mathbf{v}^{\downarrow Pa(C)}\right) \cdot U_{\text{old}}\left(c, \mathbf{v}^{\downarrow Pa(U)\setminus\{C\}}\right), \quad (3)$$

where $\mathbf{V} = Pa(C) \cup (Pa(U)\setminus\{C\})$ and \downarrow denotes the projection of configurations; for example, $\mathbf{v}^{\downarrow Pa(C)}$ is the configuration of $Pa(C)$ in which every parent of C takes the same value as in \mathbf{v} .

3. **Decision node removal.** A decision node D whose only child is the utility node U can be removed from the ID by drawing links from each parent of D to U . If $U_{\text{old}}(pa(U))$ is the utility potential before eliminating D , the new utility potential after eliminating D is

$$U_{\text{new}}(\mathbf{v}) = \max_d U_{\text{old}}\left(d, \mathbf{v}^{\downarrow Pa(U)\setminus\{D\}}\right), \quad (4)$$

where $\mathbf{V} = Pa(D) \cup (Pa(U)\setminus\{D\})$. The values of d that maximize the expected utility for each configuration \mathbf{v} determine the optimal policy for D .

4. **Arc reversal.** If the ID contains a link $X \rightarrow Y$ such that there is no other directed path from X to Y in the graph, this link can be reversed, i.e., replaced by $Y \rightarrow X$, by performing the following additional operations: (1) all the parents of X become parents of Y , and vice versa; (2) the new probability of Y is

$$P_{\text{new}}(y|\mathbf{a}, \mathbf{b}, \mathbf{c}) = \sum_x P_{\text{old}}(x|\mathbf{a}, \mathbf{b}) \cdot P_{\text{old}}(y|\mathbf{b}, \mathbf{c}), \quad (5)$$

and the new probability of X is

$$P_{\text{new}}(x|y, \mathbf{a}, \mathbf{b}, \mathbf{c}) = \frac{P_{\text{old}}(x|\mathbf{a}, \mathbf{b}) \cdot P_{\text{old}}(y|\mathbf{b}, \mathbf{c})}{P_{\text{new}}(y|\mathbf{a}, \mathbf{b}, \mathbf{c})}, \quad (6)$$

where $\mathbf{A} = Pa(X)\setminus Pa(Y)$, $\mathbf{B} = Pa(X) \cap Pa(Y)$, and $\mathbf{C} = Pa(Y)\setminus\{Pa(X) \cup X\}$.

Each of these operations transforms an ID into an equivalent ID having the same expected utility and the same optimal policies for the remaining decisions.

If the ID contains a chance node C having no decision as a child, that node must be removed before any decision. If C is the parent of other chance nodes, there is a sequence of arc reversals that generate an equivalent ID in which either C is a barren node or U is the only child of C —see Theorem 4 in [12]. Therefore, it is always possible to remove a chance or a decision node, until only U remains; at the end the potential for U contains a single real number, which is the expected utility of the ID.

A.2 Cost-effectiveness partition

Definition 1 A cost-effectiveness partition of n intervals is a tuple $Q = (\Theta_Q, C_Q, E_Q, I_Q)$, where:

- $\Theta_Q = \{\theta_1, \dots, \theta_{n-1}\}$ is a set of $n - 1$ values (thresholds), such that $0 < \theta_1 < \dots < \theta_{n-1}$,
- $C_Q = \{c_0, \dots, c_{n-1}\}$ is a set of n values (costs),
- $E_Q = \{e_0, \dots, e_{n-1}\}$ is a set of n effectiveness values, and
- $I_Q = \{I_0, \dots, I_{n-1}\}$ is a set of n interventions.

For the sake of coherence, in this presentation we establish that $\theta_0 = 0$ and $\theta_n = +\infty$ for every CEP.

Alternatively, a CEP can be denoted by a set of n 4-tuples of the form (interval, cost, effectiveness, intervention),

$$Q = \{((0, \theta_1), c_0, e_0, I_0), \\ ((\theta_1, \theta_2), c_1, e_1, I_1), \\ \dots, \\ ((\theta_{n-1}, +\infty), c_{n-1}, e_{n-1}, I_{n-1})\},$$

which means that when λ is in the interval (θ_i, θ_{i+1}) the most beneficial intervention is I_i , which has a cost c_i and an effectiveness e_i , as shown in Figure 1. When $\lambda = \theta_{i+1}$, there is a tie between I_i and I_{i+1} .

Put formally, we may define the function *index*:

$$index_Q(\lambda) = \max\{i \mid \theta_i \leq \lambda\} \quad (7)$$

and the following three functions:

$$cost_Q(\lambda) = c_i \quad (8)$$

$$eff_Q(\lambda) = e_i \quad (9)$$

$$interv_Q(\lambda) = I_i, \quad (10)$$

where $i = index_Q(\lambda)$. The net monetary benefit for a particular value of λ is

$$NMB_Q(\lambda) = \lambda \cdot eff_Q(\lambda) - cost_Q(\lambda) \quad (11)$$

It is easy to see that if two values of λ lie in the same subinterval of a partition, then they have the same cost and the same effectiveness.

A.3 Algorithms for deterministic CEA

When an intervention I_i is more effective and less expensive than I_j , the former always has a higher monetary benefit than the latter (cf. Eq. 1); we then say that I_i dominates I_j (simple dominance). When I_i is more effective and more expensive than I_j , we can define the incremental cost-effectiveness ratio:

$$ICER(I_j, I_i) = \frac{c_i - c_j}{e_i - e_j}.$$

and conclude that

$$\lambda < ICER(I_j, I_i) \implies NMB_i(\lambda) > NMB_j(\lambda) \quad (12)$$

$$\lambda > ICER(I_j, I_i) \implies NMB_j(\lambda) > NMB_i(\lambda). \quad (13)$$

Therefore the ICER separates two intervals of λ having different optimal interventions and, consequently, different cost and different effectiveness.

The standard algorithm for CEA analysis eliminates first the interventions dominated by another intervention (simple dominance), then those dominated by a pair of other interventions (extended dominance), and finally computes the ICERs—see [14] or any book on medical decision analysis.

An alternative is Algorithm 1 [6], which selects first the intervention with the lowest cost and at each step of the while loop it selects the intervention having the smallest ICER with respect to the previously selected intervention. This algorithm is slightly more efficient because it requires fewer comparisons of pairs of interventions and no comparison involving three interventions.

Algorithm 1: Deterministic CEA

Input: A set $I = \{I_1, \dots, I_m\}$ of interventions.
Result: A CEP $Q = (\Theta_Q, C_Q, E_Q, I_Q)$, with $I_Q = \{I_{\sigma(0)}, \dots, I_{\sigma(n-1)}\}$.

- 1 $\sigma(0) := \arg \min_i \text{cost}(I_i)$
- 2 $R_0 := \{i \mid I_i \in I \wedge \text{eff}(I_i) > \text{eff}(I_{\sigma(0)})\}$
- 3 $i := 1;$
- 4 **while** $R_{i-1} \neq \emptyset$ **do**
- 5 $\sigma(i) := \arg \min_{j \in R_i} \text{ICER}(I_{\sigma(i-1)}, I_j)$
- 6 $\theta_i := \text{ICER}(I_{\sigma(i-1)}, I_{\sigma(i)})$
- 7 $c_i := \text{cost}(I_{\sigma(i)})$
- 8 $e_i := \text{eff}(I_{\sigma(i)})$
- 9 $R_i := \{j \mid j \in R_{i-1} \wedge \text{eff}(I_j) > \text{eff}(I_{\sigma(i)})\}$
- 10 $i := i + 1$

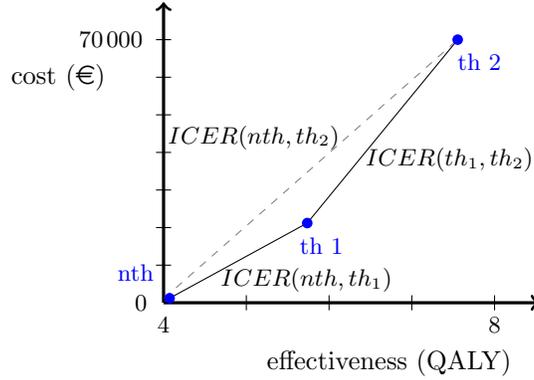


Figure 3: The result of applying Algorithm 1 to the interventions $\{nth, th1, th2\}$ is a CEP whose thresholds are the slopes of the lines from nth to $th1$ and from $th1$ to $th2$.

Example 2 Let us consider three interventions, $\{nth = \text{no therapy}, th1 = \text{therapy 1}, th2 = \text{therapy 2}\}$, whose costs are $\{150, 20\,150, 70\,150\}$ and whose values of effectiveness are $\{4.05, 5.91, 7.41\}$, respectively, as shown in Figure 3.

Algorithm 1 begins by selecting the intervention having the lowest cost, which is nth . Therefore, $\sigma_0 = 0$ and $R_0 = \{1, 2\}$. Then,⁵

$$\text{ICER}(nth, th1) = \frac{c_1 - c_0}{e_1 - e_0} = 10\,739 \text{ €/QALY},$$

which corresponds to the slope of the line nth – $th1$ in Figure 3. Similarly, $\text{ICER}(nth, th2) = 20\,832 \text{ €/QALY}$, which corresponds to the slope of the dotted line in Figure 3. Given that $\text{ICER}(nth, th1) < \text{ICER}(nth, th2)$, we have $\sigma(1) = 1$, and the first threshold is $\theta_1 = 10\,739 \text{ €/QALY}$.

In the second iteration of the “for” loop we obtain $\theta_1 = \text{ICER}(th1, th2) = 33\,384 \text{ €/QALY}$. Therefore, the resulting CEP is $\{((0, 10\,739), 150, 4.05, \text{“no therapy”}), ((10\,739, 33\,384), 20\,150, 5.91, \text{“therapy 1”}), ((33\,384, +\infty), 70\,150, 7.41, \text{“therapy 2”})\}$.

A.4 Operations with CEPs

In this section we describe the basic operations for evaluating cost-effectiveness IDs. The weighted average of CEPs generalizes the weighted average of utilities and, analogously, finding the optimal CEP generalizes the maximization of a set of utilities.

⁵There is an apparent inaccuracy in the numerical results because in the arithmetic operations we have used the values obtained from the evaluation of Example 1 (Table 3), while in the statement of the example we have rounded the values.

A.4.1 Weighted average

Definition 2 (Weighted average of CEPs) Given a set of m CEPs $\{Q_1, \dots, Q_m\}$, a chance variable X , whose domain is $\{x_1, \dots, x_m\}$, and a probability distribution for X , $P(x_j)$, we say that a CEP Q is a weighted average of the CEPs if

$$\forall \lambda, \quad \text{cost}_Q(\lambda) = \sum_{j=1}^m P(x_j) \cdot \text{cost}_{Q_j}(\lambda) \quad (14)$$

and

$$\forall \lambda, \quad \text{eff}_Q(\lambda) = \sum_{j=1}^m P(x_j) \cdot \text{eff}_{Q_j}(\lambda). \quad (15)$$

A straightforward consequence of this definition is that, because of Equation 1,

$$\forall \lambda, \quad \text{NMB}_Q(\lambda) = \sum_{j=1}^m P(x_j) \cdot \text{NMB}_{Q_j}(\lambda). \quad (16)$$

These three equalities mean that for every value of λ , the cost, effectiveness, and NMB of the weighted average partition Q are the same as if we had performed a weighted average for the values of cost and effectiveness of the Q_j 's.

The partition Q can be efficiently computed by Algorithm 2, which collects all the thresholds of the Q_j 's (line 3) and averages the costs and the effectiveness values inside each interval (lines 6 and 7). If for a certain interval all the Q_j 's have the same intervention, then this is the intervention for Q (line 9); otherwise, the optimal intervention for Q depends on the value of X : “if the chance variable X takes on the value x_j , then follow the policy indicated by the i -th partition, Q_i ” (line 11).

The reason for removing the Q_j 's such that $P(x_j) = 0$ (line 1) is that those CEPs do not affect the average values computed at lines 6 and 7, and the intervention generated at line 9 should not include the clause “if $X = x_j$ ” when this condition is never met.⁶

Algorithm 2: Weighted average of CEPs

Input: A set of m CEPs $\{Q_1, \dots, Q_m\}$, with $Q_j = (\Theta_j, C_j, E_j, I_j)$, a chance variable X , whose domain is $\{x_1, \dots, x_m\}$, and a probability distribution for X , $P(x_j)$.

Result: A new CEP $Q = (\Theta, C, E, I)$.

```

1 remove the  $Q_j$ 's such that  $P(x_j) = 0$ ;
2  $m \leftarrow$  number of remaining  $Q_j$ 's;
3  $\Theta \leftarrow \bigcup_{j=1}^m \Theta_j$ ;
4  $n \leftarrow \text{card}(\Theta)$ ;
5 for  $i \leftarrow 1$  to  $n$  do
6    $c_i \leftarrow \sum_{j=1}^m P(x_j) \cdot \text{cost}_{Q_j}(\theta_i)$ ;
7    $e_i \leftarrow \sum_{j=1}^m P(x_j) \cdot \text{eff}_{Q_j}(\theta_i)$ ;
8   if  $\text{interv}_{Q_j}(\theta_i) = \text{interv}_{Q_1}(\theta_i)$  for all  $j$  then
9      $I_i \leftarrow \text{interv}_{Q_1}(\theta_i)$ 
10  else
11   $I_i \leftarrow$  “if  $X = x_1$ , then  $\text{interv}_{Q_1}(\theta_i)$ ; if  $X = x_2$ , then  $\text{interv}_{Q_2}(\theta_i)$ ...”

```

Example 3 Let us consider two CEPs ($m = 2$), $Q_1 = \{((0, 10739), 150, 4.05, \text{“no therapy”}), ((10739, 33384), 20150, 5.91, \text{“therapy 1”}), ((33384, +\infty), 70150, 7.41, \text{“therapy 2”})\}$ and $Q_2 = \{((0, +\infty), 150, 9.85, \text{“no therapy”})\}$, a chance variable $X = \text{Test}$, whose domain is $\{\text{positive}, \text{negative}\}$, and a probability distribution $\{P(\text{positive}) = 0.186, P(\text{negative}) = 0.814\}$.

⁶If there is a j such that $P(x_j) = 1$, then $m = 1$ at line 2 and consequently $Q = Q_j$. Therefore, checking whether $m = 1$ before executing the rest of the algorithm might avoid some computations. However, we have not included this optimization in the pseudo-code of the algorithm to keep it clearer.

When applying Algorithm 2, Q_1 contributes two thresholds: $\Theta_1 = \{10739, 33384\}$; Q_2 does not contribute any: $\Theta_2 = \emptyset$. Therefore, $\Theta = \Theta_1 \cup \Theta_2 = \{10739, 33384\}$. These two thresholds define three intervals ($n = 3$). In each interval, the cost and the effectiveness will be the weighted average of the corresponding values in the input CEPs. For example, in the first interval.

$$c_1 = 0.186 \times 150 + 0.814 \times 150 = 150 \text{ €}$$

$$e_1 = 0.186 \times 4.05 + 0.814 \times 9.85 = 8.77 \text{ QALY}.$$

In this interval, the optimal intervention is “no therapy” for both results of the test; therefore, the resulting intervention is “no therapy” (cf. line 9 of Algorithm 2). In contrast, in the interval $(10739, 33384)$ the optimal intervention depends on the result of the test; the optimal intervention generated at line 11 of Algorithm 2 is “if Test = positive, then therapy 1; if Test = negative, then no therapy”. Finally, the weighted average of these CEPs is the partition $\{((0, 10739), 150, 8.77, \text{“no therapy”}), ((10739, 33384), 3874, 9.11, \text{“if Test = positive, then therapy 1; if Test = negative, then no therapy”}), ((33384, +\infty), 13184, 9.39, \text{“if Test = positive, then therapy 2; if Test = negative, then no therapy”})\}$.

Algorithm 2 is a generalization of Equation 3: just as that equation computes the weighted average of utilities when removing a chance node from a unicriterion ID, the removal of a chance node from a cost-effectiveness ID generates a new utility potential, U_{new} , such that each CEP $U_{\text{new}}(\mathbf{v})$ is computed with Algorithm 2 taking $P(c|\mathbf{v}^{\downarrow Pa(C)})$ as the input probability distribution and the set $U_{\text{old}}(c, \mathbf{v}^{\downarrow Pa(U) \setminus \{C\}})$ —for all the values of C —as the input CEPs.

A.4.2 Optimal CEP

Definition 3 (Optimal CEP) Given a set of m CEPs $\{Q_1, \dots, Q_m\}$ and a decision D , whose domain is $\{d_1, \dots, d_m\}$, a CEP Q is optimal if

$$\forall \lambda, \exists j, NMB_{\text{interv}_{Q_j}(\lambda)}(\lambda) = \max_{j'} NMB_{\text{interv}_{Q_{j'}}(\lambda)}(\lambda), \quad (17)$$

$$\text{interv}_Q(\lambda) = \text{“[choose option] } d_j; \text{ [then apply] } \text{interv}_{Q_j}(\lambda)\text{”}, \quad (18)$$

$$\text{cost}_Q(\lambda) = \text{cost}_{Q_j}(\lambda), \quad (19)$$

$$\text{eff}_Q(\lambda) = \text{eff}_{Q_j}(\lambda). \quad (20)$$

In this definition each CEP Q_j is associated with a particular value d_j (a possible choice) of decision D . Equation 17 means that when making decision D for a particular value of λ we select j such that $\text{interv}_{Q_j}(\lambda)$ is the intervention that attains the highest NMB. Put another way, for every λ the NMB of the optimal intervention in Q is the same as if we had performed a unicriterion maximization of the NMB for each single value of λ . Equation 18 means that the optimal intervention for decision D is to choose first the option d_j and then apply the intervention $\text{interv}_{Q_j}(\lambda)$. Obviously, the cost and effectiveness associated with intervention $\text{interv}_Q(\lambda)$ are those of Q_j (Eqs. 19 and 20).

The optimal CEP can be obtained by applying Algorithm 3. This algorithm collects all the thresholds of the Q_j 's (line 1) and performs a deterministic CEA (cf. Sec. 2.1) on each interval, which may generate new thresholds and new intervals. Finally, it eliminates the unnecessary thresholds, i.e., those that separate intervals having the same optimal intervention, the same cost, and the same effectiveness (line 7).

Example 4 Let us consider a decision $D = \text{Do test}$, whose domain is $\{\text{do test}, \text{do not test}\}$, and two CEPs ($m = 2$), $Q_1 = \{((0, 10739), 150, 8.77, \text{no therapy}), ((10739, 33384), 3874, 9.11, \text{“if Test = positive, then therapy 1; if Test = negative, then no therapy”}), ((33384, +\infty), 13184, 9.39, \text{“if Test = positive, then therapy 2; if Test = negative, then no therapy”})\}$ and $Q_2 = \{((0, 65360), 0, 8.77, \text{“no therapy”}), ((65360, +\infty), 20000, 9.07, \text{“therapy 1”})\}$.

When applying Algorithm 3, Q_1 contributes two thresholds: $\Theta_1 = \{10739, 33384\}$; Q_2 contributes one: $\Theta_2 = \{65360\}$. Therefore, $\Theta = \Theta_1 \cup \Theta_2 = \{10739, 33384, 65360\}$. These three thresholds define four intervals ($n = 4$). In each interval, we perform a deterministic CEA with Algorithm 1 (as we did in Example 2) considering two interventions, one for “do test” and another one for “do not test”.

In the first interval, “do not test” dominates “do test”; therefore, no new threshold is generated. According to Equation 18, the optimal policy for this interval is “do not test; no therapy”. In the second interval, the ICER is $(3874 - 0)/(9.11 - 8.77) = 11171 \text{ €}/\text{QALY}$, which becomes a new threshold. In the third interval, $(33384, 65360)$, the ICER is $(13184 - 0)/(9.39 - 8.77) = 21072 \text{ €}/\text{QALY}$, but this

Algorithm 3: Optimal CEP

Input: A variable D whose domain is $\{d_1, \dots, d_m\}$ and
a set of m CEPs $\{Q_1, \dots, Q_m\}$, with $Q_j = (\Theta_j, C_j, E_j, I_j)$

Result: A new CEP $Q = (\Theta, C, E, I)$.

- 1 $\Theta \leftarrow \bigcup_{j=1}^m \theta_j$;
 - 2 $n \leftarrow \text{card}(\Theta)$;
 - 3 **for** $i \leftarrow 1$ **to** n **do**
 - 4 perform a deterministic CEA analysis on the i -th interval;
 - 5 add the new thresholds to Θ ;
 - 6 assign the cost, effectiveness, and optimal intervention of each interval (Eqs. 18 to 20);
 - 7 remove the unnecessary thresholds.
-

threshold is not added to Θ because it falls outside the interval under analysis. Finally, in the fourth interval “do test” dominates “do not test”, and no threshold is created. Hence, we would obtain a five-interval CEP: $\{((0, 10739), 0, 8.77, \text{“do not test; no therapy”}), ((10739, 11171), 0, 8.77, \text{“do not test; no therapy”}), ((11171, 33384), 3874, 9.11, \text{“do test; if Test = positive, then therapy 1; if Test = negative, then no therapy”}), ((33384, 65360), 13184, 9.39, \text{“do test; if Test = positive, then therapy 2; if Test = negative, then no therapy”}), ((65360, +\infty), 13184, 9.39, \text{“do test, if Test = positive, then therapy 2; if Test = negative, then no therapy”})\}$.

However, the two intervals separated by the threshold 10739 have the same cost, effectiveness, and intervention, which implies that this threshold should be removed, joining the contiguous intervals (line 7 in Algorithm 3). The same occurs for the threshold 65360. Consequently, the CEP returned by this algorithm has three intervals: $\{((0, 11171), 0, 8.77, \text{“do not test; no therapy”}), ((11171, 33384), 3874, 9.11, \text{“do test; if Test = positive, then therapy 1; if Test = negative, then no therapy”}), ((33384, +\infty), 13184, 9.39, \text{“do test; if Test = positive, then therapy 2; if Test = negative, then no therapy”})\}$.

Algorithm 3 is a generalization of Equation 4: just as that equation computes the new expected utility when removing a decision node from a unicriterion ID, the removal of a decision node from a cost-effectiveness ID generates a new utility potential, U_{new} , such that each CEP $U_{\text{new}}(\mathbf{v})$ is computed with Algorithm 3 taking the set $U_{\text{old}}(d, \mathbf{v}^{\downarrow Pa(D) \setminus \{D\}})$ —for all the values of D —as the input CEPs.

A.5 Evaluation of cost-effectiveness IDs

Now we have all the elements necessary to solve the cost-effectiveness problem stated in Example 1. As mentioned earlier, we build first the ID shown in Figure 2.⁷ It has a utility table for each utility node, as if it were a unicriterion ID; the difference is that it has two types of utility nodes.

The first step performed by the algorithm is to combine all the utility nodes into a single one, U , whose parents are $\mathbf{X} = \{Disease, Dec:Test, Therapy\}$, as explained in Section 3.2.1—see Figure 4. In this new ID, U does not have a utility table but the CEP potential shown in Table 2. In this potential, every CEP $U(\mathbf{x})$ consists of a single interval, because no threshold has been generated so far; the cost is the sum of the costs in the original ID for configuration \mathbf{x} and the effectiveness is the sum of the effectiveness values, taken from Table 1.

Given that the node *Disease* is not the parent of any decision, it must be removed first. Initially it is not possible because this node has a child, *Test*. Therefore, the algorithm reverses the arc $Disease \rightarrow Test$, thus generating the ID shown in Figure 5. In the new ID the conditional probabilities of the chance nodes are:

$$P_{\text{new}}(test | dec:test) = \sum_{disease} P_{\text{old}}(disease) \cdot P_{\text{old}}(test | disease, dec:test)$$

$$P_{\text{new}}(disease | dec:test, test) = \frac{P_{\text{old}}(disease) \cdot P_{\text{old}}(test | disease, dec:test)}{P_{\text{new}}(test | dec:test)}.$$

⁷This ID, built with OpenMarkov’s graphical user interface, is available at www.openmarkov.org/cea.

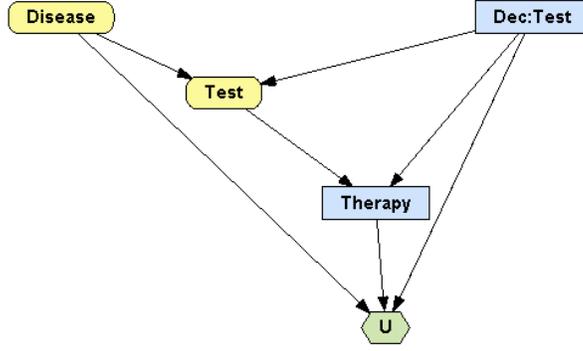


Figure 4: Initial influence diagram built by the evaluation algorithm. Node U results from combining the three utility nodes in Figure 2.

Disease	Dec:Test	Therapy	CEP
present	do test	no therapy	{ ((0, +∞), 150, 1.2, "") }
present	do test	therapy 1	{ ((0, +∞), 20150, 4.0, "") }
present	do test	therapy 2	{ ((0, +∞), 70150, 6.5, "") }
present	do not test	no therapy	{ ((0, +∞), 0, 1.2, "") }
present	do not test	therapy 1	{ ((0, +∞), 20000, 4.0, "") }
present	do not test	therapy 2	{ ((0, +∞), 70000, 6.5, "") }
absent	do test	no therapy	{ ((0, +∞), 150, 10.0, "") }
absent	do test	therapy 1	{ ((0, +∞), 20150, 9.9, "") }
absent	do test	therapy 2	{ ((0, +∞), 70150, 9.3, "") }
absent	do not test	no therapy	{ ((0, +∞), 0, 10.0, "") }
absent	do not test	therapy 1	{ ((0, +∞), 20000, 9.9, "") }
absent	do not test	therapy 2	{ ((0, +∞), 70000, 9.3, "") }

Table 2: CEP potential for the utility node in Figure 4. The values of cost and effectiveness are obtained from Table 1.

The removal of node *Disease* makes *Test* a parent of U (Figure 6). The new utility potential depends on *Dec:Test*, *Test*, and *Therapy*, as shown in Table 3. The values of cost and effectiveness in this table are obtained by averaging each pair of potentials (lines 6 and 7 of Algorithm 2). For example, the effectiveness 4.05 in the first row is obtained as the weighted average of 1.2 and 10.0, taken from Table 2, using these weights: $P(Disease = present \mid Dec:Test = do\ test, Test = positive) = 0.677$ and $P(Disease = negative, Dec:Test = do\ test, Test = positive) = 0.323$. In this table we have omitted the configurations whose probability is 0 (for example, those for “do not test” and “positive”), because they do not affect the subsequent computations (see line 1 in Algorithm 2 and the explanation in Section A.4.1).

The elimination of the decision node *Therapy* is performed by applying Algorithm 3 on each configuration of $\{Dec:Test, Test\}$, as explained in Section A.4.2, taking the three CEPs (one for each value of *Therapy*) in Table 3. For instance, the first three CEPs in that table are the same as those in Example 2; the CEP obtained in that example becomes the first CEP in Table 4.

The elimination of the chance node *Test* is performed by applying again Algorithm 2 (cf. Sec. A.4.1). In fact, the two CEPs for *do test* in Table 4 are the same we used in Example 3 and the CEP obtained in that example is the first one in Table 5.

Finally, the elimination of *Dec:Test* is performed by applying again Algorithm 3 (cf. Sec. A.4.2) on the two CEPs in Table 5, as we did in Example 4. The optimal CEP obtained in that example is shown in Table 6.

Dec:Test	Test	Therapy	CEP
do test	positive	no therapy	{ ((0, +∞), 150, 4.05, “”) }
do test	positive	therapy 1	{ ((0, +∞), 20 150, 5.91, “”) }
do test	positive	therapy 2	{ ((0, +∞), 70 150, 7.41, “”) }
do test	negative	no therapy	{ ((0, +∞), 150, 9.85, “”) }
do test	negative	therapy 1	{ ((0, +∞), 20 150, 9.80, “”) }
do test	negative	therapy 2	{ ((0, +∞), 70 150, 9.25, “”) }
do not test	not performed	no therapy	{ ((0, +∞), 0, 8.77, “”) }
do not test	not performed	therapy 1	{ ((0, +∞), 20 000, 9.07, “”) }
do not test	not performed	therapy 2	{ ((0, +∞), 70 000, 8.91, “”) }

Table 3: CEP potential obtained after eliminating the chance node *Disease*.

Dec:Test	Test	CEP
do test	positive	{ ((0, 10 739), 150, 4.05, “no therapy”), ((10 739, 33 384), 20 150, 5.91, “therapy 1”), ((33 384, +∞), 70 150, 7.41, “therapy 2”) }
do test	negative	{ ((0, +∞), 150, 9.85, “no therapy”) }
do not test	not performed	{ ((0, 65 360), 0, 8.77, “no therapy”), ((65 360, +∞), 20 000, 9.07, “therapy 1”) }

Table 4: CEP potential obtained after eliminating the decision node *Therapy*.

Dec:Test	CEP
do test	{ ((0, 10 739), 150, 8.77, “no therapy”), ((10 739, 33 384), 3874, 9.11, “if Test=positive, then therapy 1; if Test=negative, then no therapy”), ((33 384, +∞), 13 184, 9.39, “if Test=positive, then therapy 2; if Test=negative, then no therapy”) }
do not test	{ ((0, 65 360), 0, 8.77, “no therapy”), ((65 360, +∞), 20 000, 9.07, “therapy 1”) }

Table 5: CEP potential obtained after eliminating the chance node *Test*.

CEP			
λ -interval	cost	effect.	intervention
{ ((0, 11 171)	0,	8.77,	“do not test; no therapy”),
((11 171, 33 384),	3874,	9.11,	“do test;
			if Test=positive, then therapy 1;
			if Test=negative, then no therapy”),
((33 384, +∞),	13 184,	9.39,	“do test;
			if Test=positive, then therapy 2;
			if Test=negative, then no therapy”) }

Table 6: Final CEP obtained after eliminating the decision node *Dec:Test*.

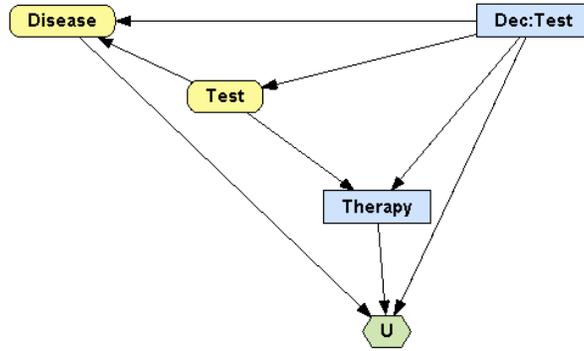


Figure 5: Influence diagram obtained after reversing the arc $Disease \rightarrow Test$.

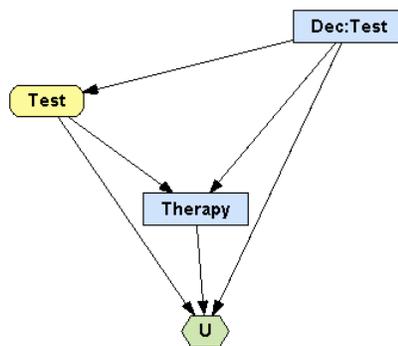


Figure 6: Influence diagram obtained after removing the node $Disease$.