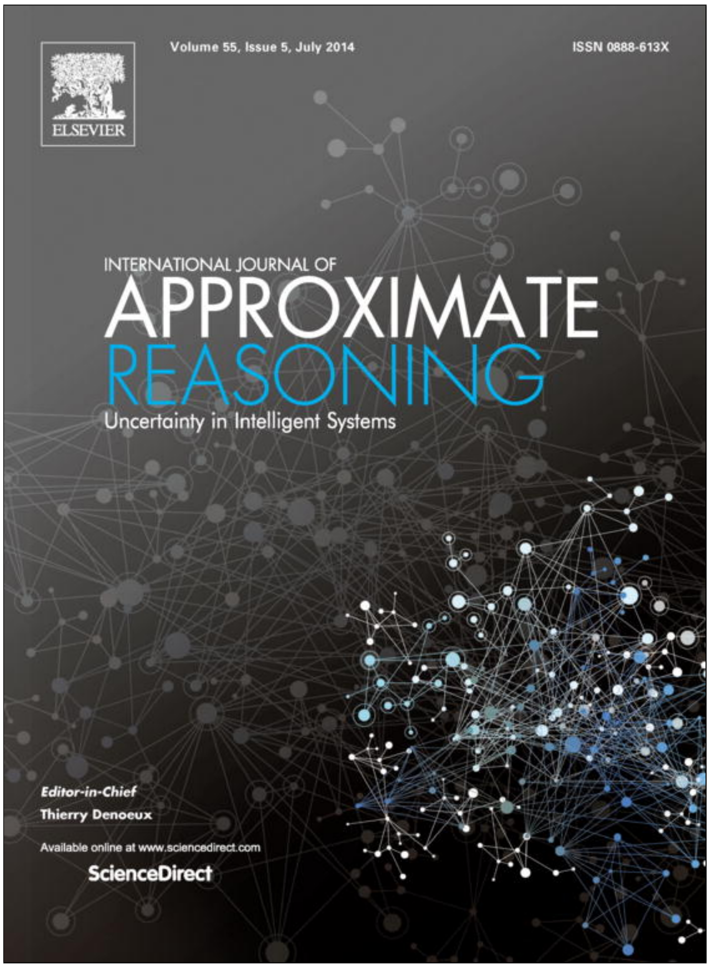
Volume 55, Issue 5, July 2014
ISSN 0888-613X
ELSEVIER
INTERNATIONAL JOURNAL OF
APPROXIMATE
REASONING
Uncertainty in Intelligent Systems
Editor-in-Chief
Thierry Denoeux
Available online at www.sciencedirect.com
ScienceDirect

International Journal of Approximate Reasoning 55 (2014) 1269–1300

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Possibilistic sequential decision making

Nahla Ben Amor [a], Hélène Fargier [b], Wided Guezguez [a]

[a] *LARODEC, Institut Supérieur de Gestion Tunis, Université de Tunis, Tunisia*
[b] *IRIT, University of Paul Sabatier, Toulouse, France*

ARTICLE INFO

Article history:
Received 21 March 2013
Received in revised form 14 November 2013
Accepted 15 November 2013
Available online 19 December 2013

Keywords:
Decision analysis
Dynamic Programming
Branch and Bound
Possibility theory
Choquet integrals
Decision tree

ABSTRACT

When the information about uncertainty cannot be quantified in a simple, probabilistic way, the topic of possibilistic decision theory is often a natural one to consider. The development of possibilistic decision theory has lead to the proposition a series of possibilistic criteria, namely: optimistic and pessimistic possibilistic qualitative criteria [7], possibilistic likely dominance [2,9], binary possibilistic utility [11] and possibilistic Choquet integrals [24]. This paper focuses on sequential decision making in possibilistic decision trees. It proposes a theoretical study on the complexity of the problem of finding an optimal strategy depending on the monotonicity property of the optimization criteria – when the criterion is transitive, this property indeed allows a polytime solving of the problem by Dynamic Programming. We show that most possibilistic decision criteria, but possibilistic Choquet integrals, satisfy monotonicity and that the corresponding optimization problems can be solved in polynomial time by Dynamic Programming. Concerning the possibilistic likely dominance criteria which is quasi-transitive but not fully transitive, we propose an extended version of Dynamic Programming which remains polynomial in the size of the decision tree. We also show that for the particular case of possibilistic Choquet integrals, the problem of finding an optimal strategy is NP-hard. It can be solved by a Branch and Bound algorithm. Experiments show that even not necessarily optimal, the strategies built by Dynamic Programming are generally very good.

1. Introduction

For several decades, there has been a growing interest in Operation Research and more recently in Artificial Intelligence towards the foundations and computational methods of decision making under uncertainty. This is especially relevant for applications to sequential decision making under uncertainty, where a suitable strategy is to be found, that associates a decision to each state of the world. Several representation formalisms can be used for sequential decision problems, such as decision trees, influence diagrams or Markov decision processes. A decision tree is an explicit representation of a sequential decision problem, while influence diagrams or Markov decision processes are compact representations. In this paper, we focus on the former framework: even in this simple, explicit, case, the set of potential strategies is combinatorial (i.e., its size increases exponentially with the size of the tree); the determination of an optimal strategy for a given representation and a given decision criterion is then an algorithmic issue in itself.

A popular criterion to compare decisions under risk is the expected utility (*EU*) model axiomatized by Von Neumann and Morgenstern [19]. This model relies on a probabilistic representation of uncertainty: an elementary decision (i.e. a one-step decision problem) is modeled by a probabilistic lottery over the possible outcomes. The preferences of the decision maker are supposed to be captured by a utility function assigning a numerical value to each outcome. The evaluation of a lottery is

E-mail addresses: nahla.benamor@gmx.fr (N. Ben Amor), fargier@irit.fr (H. Fargier), widedguezguez@gmail.com (W. Guezguez).

0888-613X/$ – see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ijar.2013.11.005

then performed through the computation of its expected utility (the greater, the better). In sequential decision making, each possible strategy is viewed as a compound lottery. It can be reduced to an equivalent simple lottery, and thus compared to remaining ones according to its expected utility.

Operational Research then proposes an efficient tool for the optimization of expected utility in probabilistic decision trees: *Dynamic Programming*. Although the high combinatorial nature of the set of possible strategies, the selection of an optimal strategy can be performed in time polynomial with the size of the decision tree: the EU model indeed satisfies a property of monotonicity that guarantees the completeness of Dynamic Programming.

When the information about uncertainty cannot be quantified in a probabilistic way the topic of possibilistic decision theory is often a natural one to consider [2,4,7]. Giving up the probabilistic quantification of uncertainty yielded to give up the EU criterion as well. The development of possibilistic decision theory has lead to the proposition and often of the characterization of a series of possibilistic counterparts of the EU criterion. Rebille [24], for instance, advocates the use of possibilistic Choquet integrals, which relies on a numerical interpretation of both possibility and utility degrees. On the contrary, Dubois and Prade [7] have studied the case of a qualitative interpretation and propose two criteria based on possibility theory, an optimistic and a pessimistic one (denoted U_{opt} and U_{pes}), whose definitions only require a finite ordinal, non-compensatory, scale for evaluating both utility and plausibility.

The axiomatization of U_{opt} and U_{pes} has given rise to the development of sophisticated qualitative models for sequential decision making, e.g. possibilistic Markov Decision Processes [25,26], possibilistic ordinal Decision Trees [10] and even possibilistic ordinal Influence Diagrams [14]. One of the most interesting properties of this qualitative model is indeed that it obeys a weak form of the monotonicity property. As a consequence, Dynamic Programming may be used and an optimal strategy with respect to U_{opt} or U_{pes} can be built in polytime, just like in the case of expected utility.

On the contrary, general Choquet integrals are incompatible with Dynamic Programming. Worst, the problem of determining an optimal strategy with respect to Choquet integrals is NP-hard in the general case [15]. We will show in the present paper that the problem of determining a strategy optimal with respect to a *possibilistic* Choquet integrals is NP-hard as well.

More generally, this paper gives a deep study of complexity of strategy optimization problem w.r.t. possibilistic decision criteria and proposes a resolution algorithm (Dynamic Programming or Branch and Bound) for each criterion according to its complexity class (P or NP).

This paper[1] is organized as follows: Section 2 presents a refresher on possibilistic decision making under uncertainty and a short survey on most common possibilistic decision criteria. Section 3 then presents our results about the complexity of sequential decision making in possibilistic decision trees. Finally, Section 4 is devoted to the proposition of a Branch and Bound algorithm for the optimization of Choquet-based possibilistic decision trees in the general case. For the sake of readability, the proofs have been gathered in Appendix A.

2. Possibilistic decision theory

2.1. Basics of possibility theory

Possibility theory, issued from Fuzzy Sets theory, was introduced by Zadeh [31] and further developed by Dubois and Prade [5]. This subsection gives some basic elements of this theory, for more details see [5].

The basic building block in possibility theory is the notion of *possibility distribution* [5]. Let $X_1, \ldots, X_n$ be a set of state variables whose value are ill-known such that $D_1, \ldots, D_n$ are their respective domains. $\Omega = D_1 \times \cdots \times D_n$ denotes the universe of discourse, which is the cartesian product of all variable domains in $X_1, \ldots, X_n$. Vectors $\omega \in \Omega$ are often called realizations or simply "states" (of the world). The agent's knowledge about the value of the x_i's can be encoded by a possibility distribution $\pi : \Omega \to [0,1]$; $\pi(\omega) = 1$ means that realization ω is totally possible and $\pi(\omega) = 0$ means that ω is an impossible state. It is generally assumed that there exist at least one state ω which is totally possible – π is said then to be *normalized*.

Extreme cases of knowledge are presented by:

- *complete knowledge*, i.e. $\exists \omega_0$ s.t. $\pi(\omega_0) = 1$ and $\forall \omega \neq \omega_0$, $\pi(\omega) = 0$,
- *total ignorance*, i.e. $\forall \omega \in \Omega$, $\pi(\omega) = 1$ (all values in Ω are possible).

From π, one can compute the possibility $\Pi(A)$ and the necessity $N(A)$ of an event $A \subseteq \Omega$:

$$\Pi(A) = \sup_{\omega \in A} \pi(\omega), \tag{1}$$

$$N(A) = 1 - \Pi(\bar{A}) = 1 - \sup_{\omega \notin A} \pi(\omega). \tag{2}$$

Measure $\Pi(A)$ evaluates to which extend A is *consistent* with the knowledge represented by π while $N(A)$ corresponds to the extent to which $\neg A$ is impossible and thus evaluates at which level A is certainly implied by the knowledge.

[1] This paper is an extended version of a preliminary work about the complexity of possibilistic decision trees presented in [8].

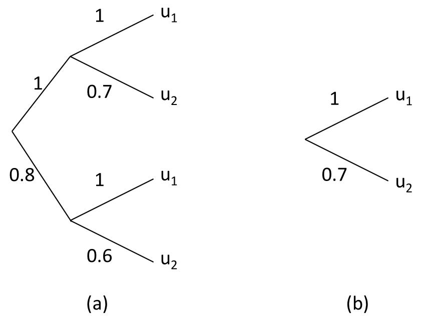

Fig. 1. A possibilistic compound lottery (a) and its reduction (b) (same reduction with $\otimes = \min$ and $\otimes = *$).

The particularity of the possibilistic scale is that it can be interpreted twofold: when the possibilistic scale is interpreted in an *ordinal* manner, i.e. when the possibility degree reflects only an ordering between the possible values, the *minimum* operator is used to combine different distributions. In a *numerical* interpretation, possibility distributions are combined using the *product* operator. In possibility theory, the possibilistic conditioning may be considered as a special case of informations fusion. It consists in revising the initial knowledge, represented by a possibility distribution π, and to change it into another possibility distribution $\pi' = \pi(.|\psi)$ with $\psi \neq \emptyset$ and $\Pi(\psi) > 0$. The two interpretations of the possibilistic scale induce two definitions of the operation of conditioning:

- min-based conditioning, which is relative to the ordinal setting:

$$\pi(\omega|\psi) = \begin{cases} 1 & \text{if } \pi(\omega) = \Pi(\psi) \text{ and } \omega \in \psi, \\ \pi(\omega) & \text{if } \pi(\omega) < \Pi(\psi) \text{ and } \omega \in \psi, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

- product-based conditioning, which is relative to the numerical setting:

$$\pi(\omega|\psi) = \begin{cases} \frac{\pi(\omega)}{\Pi(\psi)} & \text{if } \omega \in \psi, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

2.2. Possibilistic lotteries

Following [7]'s possibilistic approach of decision making under uncertainty, a decision can be seen as a possibility distribution over a finite set of outcomes. In a single stage decision making problem, a utility function maps each outcome to a utility value in a totally ordered set $U = \{u_1, \ldots, u_n\}$ (we assume without loss of generality that $u_1 \leqslant \cdots \leqslant u_n$). This function models the attractiveness of each outcome for the decision maker. An act can then be represented by a possibility distribution on U, also called a (simple) *possibilistic lottery*, and denoted by $\langle \lambda_1/u_1, \ldots, \lambda_n/u_n \rangle$: $\lambda_i = \pi(u_i)$ is the possibility that the decision leads to an outcome of utility u_i; this possibility degree are also be denoted $L[u_i]$, e.g. in the algorithms. For the sake of simplicity, we shall forget about the utility degrees that receive a possibility degree equal to 0 in a lottery. In the following, $\mathcal{L}$ denotes the set of simple lotteries (i.e. the set of possibility distributions over U).

A *possibilistic compound lottery* $L = \langle \lambda_1/L_1, \ldots, \lambda_k/L_k \rangle$ is a possibility distribution over a subset of $\mathcal{L}$. The possibility $\pi_{i,j}$ of getting a utility degree $u_j \in U$ from one of its sub-lotteries L_i depends on the possibility λ_i of getting L_i and on the conditional possibility $\lambda^i_j = \pi(u_j|L_i)$ of getting u_j from L_i, i.e. $\pi_{i,j} = \min(\lambda_j, \lambda^i_j)$ in the ordinal setting and $\pi_{i,j} = \lambda_j * \lambda^i_j$ in the numerical one. Hence, the possibility of getting u_j from a compound lottery $\langle \lambda_1/L_1, \ldots, \lambda_k/L_k \rangle$ is the *max*, over all the L_i's, of the $\pi_{i,j}$'s (and this for the two interpretation of the possibility scale). Thus, [4,7] have proposed to reduce a compound lottery $\langle \lambda_1/L_1, \ldots, \lambda_k/L_k \rangle$ (where L_i are simple lotteries) into the following equivalent simple lottery:

$$\mathit{Reduction}\big(\langle \lambda_1/L_1, \ldots, \lambda_k/L_k \rangle\big) = \Big\langle \max_{j=1..k}\big(\otimes\big(\lambda_j, \lambda^j_1\big)\big)/u_1, \ldots, \max_{j=1..k}\big(\otimes\big(\lambda_j, \lambda^j_n\big)\big)/u_n \Big\rangle \tag{5}$$

where $\otimes = \min$ for the ordinal setting and $\otimes = *$ for the numerical one.

Example 1. Let $L_1 = \langle 1/u_1, 0.7/u_2 \rangle$ and $L_2 = \langle 1/u_1, 0.6/u_2 \rangle$ be two simple lotteries, the possibilistic compound lottery $\langle 1/L_1, 0.8/L_2 \rangle$ and its reduction are represented by Fig. 1.

Obviously, the reduction of a simple lottery is the simple lottery itself. Since min and max are polynomial operations, the reduction of a compound lottery is polynomial in the size of the compound lottery.[2] We review in the following the

[2] The size of a simple lottery is the number of its outcomes; the size of a compound lottery is the sum of the sizes of its sub-lotteries plus the number of its sub-lotteries.

different criteria that have been proposed to evaluate and/or compare (simple) possibilistic lotteries; thanks to the notion of reduction, they also apply to compound possibilistic lotteries: to evaluate/compare compound lotteries, simply reduce each one to an equivalent simple one; then use one of the criteria proposed for the evaluation/the comparison of simple lotteries. Formally, any comparison criterion O, i.e. any preference order $\succcurlyeq_O$ defined on simple lotteries, can be extended to compound lotteries as follows:

$$L \succcurlyeq_O L' \iff Reduction(L) \succcurlyeq_O Reduction\big(L'\big). \tag{6}$$

Note that a lottery L is indifferent to a lottery L' w.r.t. a decision criterion O (denoted by $L \sim_O L'$) iff their reduced forms are also indifferent.

2.3. Qualitative possibilistic utilities (U_{pes}, U_{opt}, PU)

Under the assumption that the utility scale and the possibility scale are commensurate and purely ordinal, Dubois and Prade [7] have proposed the following qualitative pessimistic (denoted by U_{pes}) and optimistic (denoted by U_{opt}) utility degrees for evaluating any simple lottery $L = \langle \lambda_1/u_1, \ldots, \lambda_n/u_n \rangle$ (possibly issued from the reduction of a compound lottery):

$$U_{pes}(L) = \min_{i=1..n} \max(u_i, 1 - \lambda_i), \tag{7}$$

$$U_{opt}(L) = \max_{i=1..n} \min(u_i, \lambda_i). \tag{8}$$

U_{pes} generalizes the *Wald criterion* and estimates to what extend it is certain (i.e. necessary according to measure N) that L reaches a good utility. Its optimistic counterpart, U_{opt}, estimates to what extend it is possible that L reaches a good utility. Because decision makers are rather cautious than adventurous, the former is generally preferred to the latter. The preference order induced by U_{opt} and U_{pes} is obviously complete transitive.

Claiming that the lotteries that realize in the best prize or in the worst prize play an important role in decision making, Giang and Shenoy [11] have proposed a bipolar model in which the utility of an outcome is a pair $u = \langle \bar{u}, \underline{u} \rangle$ where $\max(\bar{u}, \underline{u}) = 1$: the utility is binary in this sense that $\bar{u}$ is interpreted as the possibility of getting the ideal, good reward (denoted $\top$) and $\underline{u}$ is interpreted as the possibility of getting the anti ideal, bad reward (denoted $\bot$).

The normalization constraint $\max(\bar{u}, \underline{u}) = 1$, implies that the set $U = \{\langle \bar{u}, \underline{u} \rangle \in [0,1]^2, \max(\bar{u}, \underline{u}) = 1\}$ is totally ordered by the relation $\succcurlyeq_{pu}$ defined by:

$$\langle \bar{u}, \underline{u} \rangle \succcurlyeq_{pu} \langle \bar{v}, \underline{v} \rangle \iff \begin{cases} \bar{u} = \bar{v} = 1 \text{ and } \underline{u} \leqslant \underline{v} \\ \quad \text{or} \\ \bar{u} \geqslant \bar{v} \text{ and } \underline{u} = \underline{v} = 1 \\ \quad \text{or} \\ \bar{u} = \underline{v} = 1 \text{ and } \bar{v} < 1. \end{cases} \tag{9}$$

From this relation we can derive that if $\bar{u} \geqslant \overline{u'}$ and $\underline{u} \leqslant \underline{u'}$, then $\langle \bar{u}, \underline{u} \rangle \succcurlyeq_{pu} \langle \bar{v}, \underline{v} \rangle$. Each $u_i = \langle \overline{u_i}, \underline{u_i} \rangle$ in the utility scale is thus understood as a small lottery $\langle \overline{u_i}/\top, \underline{u_i}/\bot \rangle$. Hence, a lottery $\langle \lambda_1/u_1, \ldots, \lambda_n/u_n \rangle$ can be viewed as a compound lottery, and its PU utility is computed by reduction:

$$\begin{aligned} &PU\big(\langle \lambda_1/u_1, \ldots, \lambda_n/u_n \rangle\big) \\ &\quad = Reduction\big(\lambda_1/\langle \overline{u_1}/\top, \underline{u_1}/\bot \rangle, \ldots, \lambda_n/\langle \overline{u_n}/\top, \underline{u_n}/\bot \rangle\big) \\ &\quad = \Big\langle \max_{j=1..n} \big(\min(\lambda_j, \overline{u_j})\big)/\top, \max_{j=1..n} \big(\min(\lambda_j, \underline{u_j})\big)/\bot \Big\rangle. \end{aligned} \tag{10}$$

We thus get, for any lottery L a binary utility $PU(L) = \langle \bar{u}, \underline{u} \rangle$ in U. Lotteries can then be compared according to Eq. (9):

$$L \succcurlyeq_{PU} L' \iff Reduction(L) \succcurlyeq_{pu} Reduction\big(L'\big). \tag{11}$$

In [13] Giang and Shenoy show that the order induced by PU is transitive and that it collapses with the one induced by U_{opt} (resp. U_{pes}) whenever for any lottery, the possibility $\underline{u}$ (resp. $\bar{u}$) of getting the worst (resp. the best) utility is equal to 1. One shall thus say that PU captures U_{opt} and U_{pes} as particular cases.

2.4. Possibilistic likely dominance ($L\Pi$, LN)

When the scales evaluating the utility and the possibility of the outcomes are not commensurate, [2,9] propose to prefer, among two possibilistic decisions, the one that is more likely to overtake the other. Such a rule does not assign a utility degree to the decisions, but draws a pairwise comparison. Although designed on a Savage-like framework rather than on lotteries, it can be translated on lotteries. This rule states that given two lotteries $L_1 = \langle \lambda_1^1/u_1^1, \ldots, \lambda_n^1/u_n^1 \rangle$ and

$L_2 = \langle \lambda_1^2/u_1^2, \ldots, \lambda_n^2/u_n^2 \rangle$, L_1 is as least as good as L_2 as soon as the likelihood (here, the necessity or the possibility) of the event "*The utility of* L_1 *is as least as good as the utility of* L_2" is greater or equal to the likelihood of the event "*The utility of* L_2 *is as least as good as the utility of* L_1". Formally:

$$L_1 \succcurlyeq_{LN} L_2 \quad \text{iff} \quad N(L_1 \succcurlyeq L_2) \geqslant N(L_2 \succcurlyeq L_1), \tag{12}$$

$$L_1 \succcurlyeq_{L\Pi} L_2 \quad \text{iff} \quad \Pi(L_1 \succcurlyeq L_2) \geqslant \Pi(L_2 \succcurlyeq L_1) \tag{13}$$

where $\Pi(L_1 \succcurlyeq L_2) = \max_{u_i^1, u_j^2 \text{s.t.}\, u_i^1 \geqslant u_j^2} (\otimes(\lambda_i^1, \lambda_j^2))$ and

$$N(L_1 \succcurlyeq L_2) = 1 - \max_{u_i^1, u_j^2 \text{ s.t. } u_i^1 < u_j^2} \big(\otimes\big(\lambda_i^1, \lambda_j^2\big)\big),$$

such that $\otimes = \min$ for ordinal setting and $\otimes = *$ for numerical setting.

The preference order induced on the lotteries is not transitive, but only quasi-transitive [2]: obviously $L_1 \succ_{LN} L_2$ and $L_2 \succ_{LN} L_3$ imply that $L_1 \succ_{LN} L_3$ (resp. $L_1 \succ_{L\Pi} L_2$ and $L_2 \succ_{L\Pi} L_3$ imply that $L_1 \succ_{L\Pi} L_3$) but it may happen that $L_1 \sim_{LN} L_2$, $L_2 \sim_{LN} L_3$ (resp. $L_1 \sim_{L\Pi} L_2$, $L_2 \sim_{L\Pi} L_3$) and $L_1 \succ_{LN} L_3$ (resp. $L_1 \succ_{L\Pi} L_3$).

Note that contrary to U_{pes}, U_{opt} and PU, which are purely ordinal, possibilistic likely dominance can be defined in the ordinal setting or the numerical setting of possibility theory.

2.5. Order of Magnitude Expected Utility (OMEU)

Order of Magnitude Expected Utility theory relies on a qualitative representation of beliefs, initially proposed by Spohn [28], via *Ordinal Conditional Functions*, and later popularized under the term *kappa-rankings*. Formally, $\kappa : 2^{\Omega} \to Z^{+} \cup \{+\infty\}$ is a kappa-ranking if and only if it obeys to the following axioms:

$$\text{(S1)} \qquad \min_{\omega \in \Omega} \kappa\big(\{\omega\}\big) = 0,$$

$$\text{(S2)} \qquad \kappa(A) = \begin{cases} \min_{\omega \in A} \kappa(\{\omega\}) & \text{if } A \neq \emptyset \text{ and } A \subseteq \Omega, \\ +\infty & \text{otherwise.} \end{cases}$$

Note that an event A is more likely than an event B if and only if $\kappa(A) < \kappa(B)$: kappa-rankings have been termed as *disbelief functions*. They receive an interpretation in terms of order of magnitude of "small" probabilities. Say "$\kappa(A) = i$" is equivalent to say that $P(A)$ is of the same order of ε^i, for a given fixed infinitesimal ε. As pointed out by [6], there exists a close link between kappa-rankings and possibility measures, insofar as any kappa-ranking can be represented by a possibility measure, and vice versa.

Order of magnitude utilities have been defined in the same way [20,29]. Namely, an order of magnitude function $\mu : X \to Z^{+} \cup \{+\infty\}$ can be defined to rank outcomes $x \in X$ in terms of degrees of "dissatisfaction". Once again, $\mu(x) < \mu(x')$ if and only if x is more desirable than x', $\mu(x) = 0$ for the most desirable consequences, and $\mu(x) = +\infty$ for the least desirable consequences: $\mu(x) = i$ is equivalent to say that the utility of x is of the same order of ε^i, for a given fixed infinitesimal ε.

An *Order of Magnitude Expected Utility* (OMEU) model can then be defined (see [20,29] among others). Considering that an order of magnitude lottery $L = \langle \kappa_1/\mu_1, \ldots, \kappa_n/\mu_n \rangle$ represents some probabilistic lottery, it is possible to compute the order of magnitude of the expected utility of this probabilistic lottery: it is equal to $\min_{i=1..n}\{\kappa_i + \mu_i\}$. Hence the definition of the OMEU value of a kappa lottery $L = \langle \kappa_1/\mu_1, \ldots, \kappa_n/\mu_n \rangle$:

$$OMEU(L) = \min_{i=1,n} \{\kappa_i + \mu_i\}. \tag{14}$$

The preference relation $\succcurlyeq_{OMEU}$ is thus defined as:

$$L \succcurlyeq_{OMEU} L' \quad \text{iff} \quad OMEU(L) \geqslant OMEU\big(L'\big). \tag{15}$$

The preference order induced on the lotteries is transitive [12].

2.6. Possibilistic Choquet integrals (Ch_N, Ch_Π)

In presence of heterogeneous information, i.e. when the knowledge about the state of the world is possibilistic while the utility degrees are numerical and compensatory, Choquet integrals [1] appear as a right way to extend expected utility to non-Bayesian models [1]. Like the EU model, this model is a numerical, compensatory, way of aggregating uncertain utilities. But it does not necessarily resort on a Bayesian modeling of uncertain knowledge. Indeed, this approach allows the use of any monotonic set function μ, also called a capacity or fuzzy measure. Such measures capture Probability measures, Necessity and Possibility measures and Belief functions, etc. as particular cases.

$$Ch_{\mu}(L) = u_1 + \sum_{i=2,n} (u_i - u_{i-1}) * \mu(L \geqslant u_i). \tag{16}$$

If μ is a probability measure then $Ch_{\mu}(L)$ is simply the expected utility of L.

Choquet integrals have been proposed by Schmeidler [27] for decision making under probabilistic ambiguity (the capacity is a lower probability) and by Quiggin [22] and Yaari [30] as a model of "rank dependent utility" (in these works, the capacity is a transformation of the original probability measure). In the present paper, we are interested in studying the possibilistic framework for decision making: for cautious (resp. adventurous) decision makers, the capacity μ is the necessity measure N (resp. the possibility measure Π):

$$Ch_N(L) = u_1 + \sum_{i=2,n} (u_i - u_{i-1}) * N(L \geqslant u_i), \tag{17}$$

$$Ch_{\Pi}(L) = u_1 + \sum_{i=2,n} (u_i - u_{i-1}) * \Pi(L \geqslant u_i). \tag{18}$$

Example 2. Let L_1 and L_2 be two possibilistic lotteries such that $L_1 = \langle 0.1/2, 0.4/4, 1/5\rangle$ and $L_2 = \langle 1/3, 0.7/5, 0.2/10\rangle$. Thus:

$$Ch_N(L_1) = 2 + \big[(4-2) * (1-0.1)\big] + \big[(5-4) * (1-0.4)\big] = 4.4,$$
$$Ch_{\Pi}(L_2) = 3 + \big[(5-3) * 0.7\big] + \big[(10-5) * 0.2\big] = 5.4.$$

The following proposition emphasizes the pessimistic character of Ch_N: increasing the possibility of a possibly, good consequence that is not better than its best possible one cannot increase (and may decrease) its evaluation.

Proposition 1. *Given a lottery $L = \langle \lambda_1/u_1, \ldots, \lambda_n/u_n\rangle$ where utilities are ranked in an increasing order. Let u_i be a utility in L s.t. $u_i \leqslant \max_{u_j \in L, \lambda_j > 0} u_j$. Let L' be a lottery constructed from L by just increasing the possibility of u_i, i.e. $\lambda'_i > \lambda_i$. Thus $Ch_N(L') \leqslant Ch_N(L)$.*

Example 3. Let L_1 and L_2 be two possibilistic lotteries such that $L = \langle 0.2/10, 0.5/20, 1/30\rangle$ and $L' = \langle 0.2/10, 1/20, 1/30\rangle$. We can check that: $Ch_N(L) = 23 > Ch_N(L') = 18$.

This result can be extended to the composition of lotteries: compounding a lottery with another lottery that is not strictly better than the original one cannot increase its Choquet value.

Proposition 2. *Let $L_1 = \langle \lambda_1^1/u_1, \ldots, \lambda_n^1/u_n\rangle$ and $L_2 = \langle \lambda_1^2/u_1, \ldots, \lambda_n^2/u_n\rangle$ be two lotteries such that*

$$\max_{u_i \in L_2, \lambda_i^2 > 0} u_i \leqslant \max_{u_i \in L_1, \lambda_i^1 > 0} u_i.$$

Then:

$$Ch_N\big(Reduction(\langle 1/L_1, 1/L_2\rangle)\big) \leqslant Ch_N(L_1).$$

So, the compound lottery $\langle 1/L_1, 1/L_2\rangle$ cannot be better (with respect to the pessimistic Choquet integral) than L_1, except when it allows to reach a utility degree not reached by L_1. Simply increasing the possibility of the good utilities reached by L_1 is not a way to increase its Ch_N utility.

3. On the complexity of decision making in possibilistic decision trees

3.1. Possibilistic decision trees

Decision trees [23] are graphical representations of sequential decision problems under the assumption of full observability. This framework proposes an explicit modeling of sequential decision problems, by representing each possible scenario by a path from the root to the leaves of the tree. Formally, the graphical component of a decision tree $\mathcal{T}$ is composed of a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$ such that the set $\mathcal{N}$ contains three kinds of nodes:

- $\mathcal{D} = \{D_0, \ldots, D_m\}$ is the set of decision nodes (represented by rectangles). The labeling of the nodes is supposed to be in accordance with the temporal order, i.e. if D_i is a descendant of D_j, then $i > j$. The root node of the tree is necessarily a decision node, denoted by D_0.
- $\mathcal{LN} = \{LN_1, \ldots, LN_k\}$ is the set of leaves, also called utility leaves: $\forall LN_i \in \mathcal{LN}$, $u(LN_i)$ is the utility of being eventually in node LN_i. For the sake of simplicity we assume that only leave nodes lead to utilities.

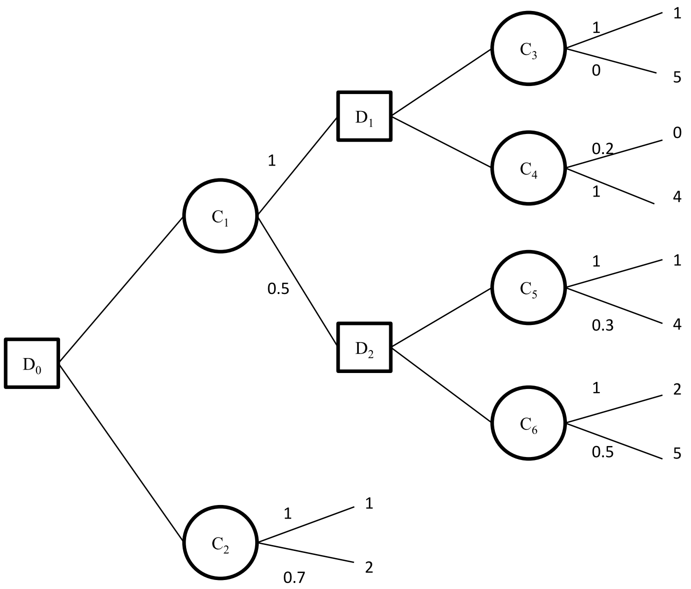

Fig. 2. Example of a possibilistic decision tree with $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5, C_6\}$, $\mathcal{D} = \{D_0, D_1, D_2\}$ and $\mathcal{LN} = U = \{0, 1, 2, 3, 4, 5\}$.

- $\mathcal{C} = \{C_1, \dots, C_n\}$ is the set of chance nodes represented by circles.
  For any $X_i \in \mathcal{N}$, $Succ(X_i) \subseteq \mathcal{N}$ denotes the set of its children. Moreover, for any $D_i \in \mathcal{D}$, $Succ(D_i) \subseteq \mathcal{C}$: $Succ(D_i)$ is the set of actions that can be decided when D_i is observed. For any $C_i \in \mathcal{C}$, $Succ(C_i) \subseteq \mathcal{LN} \cup \mathcal{D}$: $Succ(C_i)$ is indeed the set of outcomes of the action C_i – either a leaf node is observed, or a decision node is reached (and then a new action should be executed).

In the following $|\mathcal{D}|$ (resp. $|\mathcal{C}|$, $|\mathcal{LN}|$) denotes the total number of decision nodes (resp. chance nodes, leaves) in the tree. The size $|\mathcal{T}|$ of a decision tree is equal to its number of edges ($|\mathcal{E}|$) and nodes ($|\mathcal{N}|$). In classical, probabilistic, decision trees [23] the uncertainty pertaining to the possible outcomes of each $C_i \in \mathcal{C}$, is represented by a conditional probability distribution p_i on $Succ(C_i)$, such that $\forall N \in Succ(C_i)$, $p_i(N) = P(N|path(C_i))$ where $path(C_i)$ denotes all the value assignments to chance and decision nodes on the path from the root to C_i.

In the present work, we obviously use a possibilistic labeling (see Fig. 2). The difference with probabilistic decision trees is that the chance nodes are not probabilistic but possibilistic lotteries. More precisely, for any $C_i \in \mathcal{C}$, the uncertainty pertaining to the more or less possible outcomes of each C_i is represented by a *conditional possibility distribution* π_i on $Succ(C_i)$, such that $\forall N \in Succ(C_i)$, $\pi_i(N) = \Pi(N|path(C_i))$.

Solving a decision tree amounts at building a *strategy* that selects an action (i.e. a chance node) for each reachable decision node. Formally, we define a strategy as a function δ from $\mathcal{D}$ to $\mathcal{C} \cup \{\bot\}$. $\delta(D_i)$ is the action to be executed when a decision node D_i is observed. $\delta(D_i) = \bot$ means that no action has been selected for D_i (because either D_i cannot be reached or the strategy is partially defined). Admissible strategies must be:

- *sound*: $\forall D_i \in \mathcal{D}, \delta(D_i) \in Succ(D_i) \cup \{\bot\}$,
- *complete*: (i) $\delta(D_0) \neq \bot$ and (ii) $\forall D_i$ s.t. $\delta(D_i) \neq \bot, \forall N \in Succ(\delta(D_i))$, either $\delta(N) \neq \bot$ or $N \in \mathcal{LN}$.

Let Δ be the set of sound and complete strategies that can be built from the decision tree. Any strategy in Δ can be viewed as a connected subtree of the decision tree whose arcs are of the form $(D_i, \delta(D_i))$, i.e. there being exactly one decision arc left at each decision node.

The size $|\delta|$ of δ is the sum of its number of nodes and edges. It is obviously lower than the size of the decision tree (but for trivial trees that are already sound and complete strategies to start with).

Strategies can be evaluated and compared thanks to the notion of lottery reduction. Recall indeed that leaf nodes in $\mathcal{LN}$ are labeled with utility degrees. Then a chance node can be seen as a simple lottery (for the rightmost chance nodes) or as a compound lottery (for the inner chance nodes). Each strategy is a compound lottery and can be reduced to an equivalent simple one. Formally, the composition of lotteries will be applied from the leafs of the strategy to its root, according to the following recursive definition for any N_i in $\mathcal{N}$:

$$L(N_i, \delta) = \begin{cases} L(\delta(N_i), \delta) & \text{if } N_i \in \mathcal{D}, \\ Reduction(\langle \pi_i(X_j)/L(X_j, \delta)_{X_j \in Succ(N_i)} \rangle) & \text{if } N_i \in \mathcal{C}, \\ \langle 1/u(N_i) \rangle & \text{if } N_i \in \mathcal{LN}. \end{cases} \tag{19}$$

Eq. (19) is simply the adaptation to strategies of lottery reduction (Eq. (5)). We can then compute $Reduction(\delta) = L(D_0, \delta)$: $Reduction(\delta)(u_i)$ is simply the possibility of getting utility u_i when δ is applied from D_0. Since operators max and $*$ (resp. max and min) are polytime Eq. (19) defines a polytime computation of the reduced lottery.

Proposition 3. *For any strategy δ in Δ, the simple possibilistic lottery $Reduction(\delta)$ can be computed in polytime.*

We are now in position to compare strategies, and thus to define the notion of optimality. Let O be one of the decision criteria defined above (i.e. depending on the application, $\succeq_O$ is either $\succeq_{L\Pi}$, or $\succeq_{LN}$, or the order induced by U_{pes}, or by U_{opt}, etc.). A strategy $\delta \in \Delta$, is said to be optimal w.r.t. $\succeq_O$ iff:

$$\forall \delta' \in \Delta, \quad Reduction(\delta) \succeq_O Reduction\big(\delta'\big). \tag{20}$$

Note that this definition does not require the full transitivity of $\succeq_O$ and is meaningful as soon as $\succeq_O$ is complete and its the strict part, i.e. $\succ_O$, is transitive. This means that it is applicable to the preference relations that rely on the comparison of utility degrees (qualitative utilities, binary utility, Choquet integrals) but also to $\succeq_{LN}$ and $\succeq_{L\Pi}$.

3.2. Dynamic Programming in possibilistic decision trees

Finding optimal strategies via an exhaustive enumeration of δ is a highly computational task. For instance, in a decision tree with n decision nodes and a branching factor equal to 2, the number of potential strategies is in $O(2^{\sqrt{n}})$.

For standard probabilistic decision trees, where the goal is to maximize expected utility, an optimal strategy can be computed in polytime (with respect to the size of the tree) thanks to an algorithm of Dynamic Programming which builds the best strategy backwards, optimizing the decisions from the leaves of the tree to its root. Note that this algorithm does not generate all the best strategies but returns only one among them.

Regarding qualitative decision trees, Garcia and Sabbadin [10] have shown that such a method can also be used to get a strategy maximizing U_{pes} and U_{opt}. The reason is that, like EU, U_{pes} and U_{opt} satisfy the key properties of transitivity and weak monotonicity.

Formally a preference order $\succeq_O$ is said to be weakly monotonic iff whatever L, L' and L'', whatever (α, β) such that $\max(\alpha, \beta) = 1$:

$$L \succeq_O L' \quad \Rightarrow \quad \big\langle \alpha/L, \beta/L'' \big\rangle \succeq_O \big\langle \alpha/L', \beta/L'' \big\rangle. \tag{21}$$

For the sake of simplicity, we write here the property for a binary composition of lotteries. By associativity of possibilistic composition, it obviously applies to a composition of more than two lotteries.

This property states that the combination of L (resp. L') with L'', does not change the initial order induced by O between L and L' – this allows Dynamic Programming to decide in favor of L or L' before considering the compound decision. The principle of backwards reasoning procedure is depicted in a recursive manner by Algorithm 1 for a preference order $\succeq_O$ over possibilistic lotteries.

In this algorithm $L[u_i]$ is the possibility degree to have the utility u_i in the lottery L, $Succ(N).first$ is the first node in the set of successors of N, i.e. $Succ(N)$ and $\otimes$ (Line 1) is the min operator for the ordinal setting and the product operator for the numerical setting.

The principle of this algorithm is as follows: When each chance node is reached, an optimal sub-strategy is built for each of its children – these sub-strategies are combined w.r.t. their possibility degrees, and the resulting compound strategy is reduced: we get an equivalent simple lottery, representing the current optimal sub-strategy. When a decision node X is reached, a decision Y^* leading to a sub-strategy optimal w.r.t. $\succeq_O$ is selected among all the possible decisions $Y \in Succ(X)$, by comparing the simple lotteries equivalent to each sub-strategy.

Algorithm 1, Line 2 performs the comparison of two simple lotteries according to the criterion O to optimize (e.g. $Ch_N(L_Y)$ and $Ch_N(L_Y*)$) This procedure crosses each edge in the tree only once. When the comparison of simple lotteries by $\succeq_O$ (Algorithm 1, Line 2) and the reduction operation on a 2-level lottery (Algorithm 1, Line 1) can be performed in polytime, its complexity is polynomial w.r.t. the size of the tree.

Proposition 4. *If $\succeq_O$ is transitive and satisfies the weak monotonicity property, then Algorithm 1 computes a strategy optimal w.r.t. criterion O in time polynomial with respect to the size of the decision tree (assuming that the reduction operation and the comparison of lotteries can be performed in polytime).*

We will see in the following that, beyond U_{pes} and U_{opt} criteria, several other criteria satisfy the monotonicity property and that their optimization can be managed in polytime by Dynamic Programming. The possibilistic Choquet integrals, on the contrary, do not satisfy weak monotonicity; we show in the next section that they lead to NP-complete decision problems.

```
Algorithm 1: ProgDyn($N$: node, $\delta$: strategy).
            Data: $O$ is the criterion to optimize.
            Result: A lottery $L$
            begin
                // Initialization
                for $i \in \{1,\dots,n\}$ do $L[u_i] \leftarrow 0$ // Leaves
                if $N \in \mathcal{LN}$ then $L[u(N)] \leftarrow 1$ // Chance nodes
                if $N \in \mathcal{C}$ then
                    // Reduce the compound lottery
                    foreach $Y \in Succ(N)$ do
                        $L_Y \leftarrow ProgDyn(Y,\delta)$;
                        for $i \in \{1,\dots,n\}$ do
(Line 1)                    $L[u_i] \leftarrow \max(L[u_i], (\otimes(\pi_N(Y), L_Y[u_i])))$;

                // Decision nodes
                if $N \in \mathcal{D}$ then
                    // Choose the best decision
                    $Y^* \leftarrow Succ(N).first$;
                    foreach $Y \in Succ(N)$ do
                        $L_Y \leftarrow ProgDyn(Y,\delta)$;
(Line 2)                if $L_Y \succ_O L_{Y^*}$ then $Y^* \leftarrow Y$
                    $\delta(N) \leftarrow Y^*$;
                    $L \leftarrow L_{Y^*}$;
                return $L$;
```

3.3. On the complexity of possibilistic decision trees

Formally, for any of the optimization criteria proposed above, the corresponding decision problem can be defined as follows:

Definition 1 *(DT-OPT-O).* (Strategy optimization w.r.t. an optimization criterion O in possibilistic decision trees.)

INSTANCE: A possibilistic decision tree $\mathcal{T}$, a constant lottery θ.
QUESTION: Does there exist a strategy $\delta \in \Delta$ such that *Reduction*$(\delta) \succeq_O \theta$*?*

For instance DT-OPT-Ch_N corresponds to the optimization of the necessity-based Choquet integrals. DT-OPT-U_{pes} and DT-OPT-U_{opt} correspond to the optimization of the possibilistic qualitative utilities U_{pes} and U_{opt}, respectively.

3.3.1. Possibilistic Qualitative Utilities (U_{pes}, U_{opt}, PU)

Possibilistic qualitative utilities U_{pes} and U_{opt} satisfy the weak monotonicity principle. Although not referring to a classical, real-valued utility scale, but to a 2 dimensional scale, this is also true in the case of *PU*.

Proposition 5. *$\succeq_{U_{pes}}$, $\succeq_{U_{opt}}$ and $\succeq_{PU}$ satisfy the weak monotonicity property for both $\otimes = \min$ and $\otimes = *$.*

This proposition is not explicitly proved in the literature although it is a common knowledge in qualitative possibilistic decision theory (see [7,11]); it is also known that Dynamic Programming applies to the optimization of U_{pes}, U_{opt} and PU in qualitative possibilistic Markov decision processes and thus to decision trees (see [10,21,26]). We prove it in the general case – i.e. for both the qualitative case ($\otimes = \min$) and the quantitative case ($\otimes = *$).

Dynamic Programming applies to the optimization of U_{pes}, U_{opt} and PU: the polynomiality of these criteria is a direct consequence of Propositions 4 and 5.

Corollary 1. *DT-OPT-U_{pes}, DT-OPT-U_{opt} and DT-OPT-PU belong to P for both $\otimes = \min$ and $\otimes = *$.*

3.3.2. Possibilistic Likely Dominance ($L\Pi$, LN)

a) *Comparing lotteries by $L\Pi$* Although $\succeq_{L\Pi}$ is not a fully transitive criterion (it is only quasi-transitive). It is easy to show that:

Proposition 6. *$\succeq_{L\Pi}$ satisfies the weak monotonicity principle.*

Because this criterion is not fully transitive but only quasi-transitive, classical Dynamic Programming shouldn't be applied. Nevertheless, it can be extended so as to keep, for each decision node, a judicious *subset* of all the best (but possibly indifferent) strategies for the current subtree [17,18] – see Algorithm 2. The recursive procedure thus returns a *set* S of simple lotteries and their associated strategies: dominated strategies are removed by a call to a pruning function – see Algorithm 3, thanks to the monotonicity principle.

At each chance node, a pointwise composition of the lotteries returned by the recursive calls is performed: this leads to a set of reduced lotteries that is pruned so as to keep, again, judicious non-dominated decisions. The difficulty with this type of procedure it that the size of the set of lotteries handled by the algorithm may grow exponentially. We show in the following that it is possible to keep it within polynomial bounds when the criterion to optimize is $L\Pi$. Indeed, when $O = L\Pi$, the comparison of two lotteries only depends on the relative position of their cores. Formally, for any lottery L, let $L_{=\alpha} = \{u_i, L[u_i] = \alpha\}$; the core of L corresponds to $L_{=1}$. It is easy to show that $L \succcurlyeq_{L\Pi} L'$ iff $\max(L_{=1}) \geqslant \min(L'_{=1})$. This allows us to replace any simple lottery L built during the computation by a simple lottery where any utility degree different from $\max(L_{=1})$ and $\min(L_{=1})$ receives a possibility degree of 0. We call this lottery called the *core lottery* of L and denote it by L^{core}; namely:

$$L^{core} = \big\langle 1/\min(L_{=1}), 1/\max(L_{=1})\big\rangle.$$

Example 4. Let $L = \langle 0.7/10, 0.5/20, 1/30, 1/40\rangle$ and $L' = \langle 0.2/10, 1/20, 0.9/30\rangle$ be two simple lotteries. $L^{core} = \langle 1/30, 1/40\rangle$ and $L'^{core} = \langle 1/20\rangle$.

It holds, in both numerical and ordinal settings, that

$\Pi(L \succcurlyeq L') = 1$ and $\Pi(L' \succcurlyeq L) = 0.9$: $L \succcurlyeq_{L\Pi} L'$; and that

$\Pi(L^{core} \succcurlyeq L'^{core}) = 1$ and $\Pi(L'^{core} \succcurlyeq L^{core}) = 0$: $L^{core} \succcurlyeq_{L\Pi} L'^{core}$.

It can now be shown that:

Proposition 7. *$\forall L, L' \in \mathcal{L}$, $L \succcurlyeq_{L\Pi} L' \iff L^{core} \succcurlyeq_{L\Pi} L'$ and $L' \succcurlyeq_{L\Pi} L \iff L' \succcurlyeq_{L\Pi} L^{core}$.*

A first consequence of this proposition is that replacing any simple lottery by its core lottery during the computation does not change the set of optimal strategies of the problem. A second consequence is that if $L^{core} = L'^{core}$, then for any L'', $L \succcurlyeq_{L\Pi} L'' \iff L' \succcurlyeq_{L\Pi} L''$ and $L'' \succcurlyeq_{L\Pi} L \iff L'' \succcurlyeq_{L\Pi} L'$ which means that lotteries sharing the same core form an equivalence class. We shall then remove from S the pairs (L^{core}, δ) such that there exist in the set another pair (L'^{core}, δ'): since leading to the same core lottery, δ and δ' are equivalent w.r.t. $L\Pi$.

Then we can remark that the number of simple core lotteries is bounded by n^2, n being the number of different utility values labeling the leaves. As a consequence, the set S of lotteries handled by Extended Dynamic Programming (Algorithm 2) when optimizing $L\Pi$ contains at most n^2 elements. The different loops on S in this function are then also polynomially bounded: $O(n^4)$ is a trivial upper bound for the pruning function (assuming that the comparison of simple lotteries can be done in linear time and the pointwise composition of the sets is bounded by $O(m.n^4)$, m being the number of children of the chance node). The Extended Dynamic Programming procedure is thus polynomial in the size of the decision tree when $O = L\Pi$.

b) *Comparing lotteries by LN* When $O = LN$, the principle is similar to $O = L\Pi$ but a little more tricky. The idea is to replace any simple lottery L by a lottery which keeps track of the bounds of the core (as it is the case with $L\Pi$ criterion) but also of the bound of the next cut of L, since only these two levels play a role in the $\succcurlyeq_{LN}$ comparison. We call this lottery the *2nd core lottery* of L and denote it L^{2nd}. Formally, let $L_{=\alpha} = \{u_i, L[u_i] = \alpha\}$ and $\alpha^{2nd}(L) = \max_{u_i, L[u_i]<1} L[u_i]$, then:

$$L^{2nd} = \begin{cases} \langle 1/\min(L_{=1}),\ 1/\max(L_{=1})\rangle & \text{if } \max(L_{=1}) \neq \min(L_{=1}), \\ \langle 1/\max(L_{=1}),\ \alpha^{2nd}(L)/\min(L_{=\alpha^{2nd}}),\ \alpha^{2nd}(L)/\max(L_{=\alpha^{2nd}})\rangle & \text{otherwise.} \end{cases}$$

The second point corresponds to the case where L (and thus L^{2nd}) is unimodal with mode $\max(L_{=1}) = \min(L_{=1})$. In other terms, a 2nd core lottery is:

- either of the form $\langle 1/m, 1/M\rangle$ where $m = \min(L_{=1}) < M = \max(L_{=1})$,
- or of the form $\langle \alpha/m, 1/u, \alpha/M\rangle$ where $u = \max(L_{=1})$, $\alpha = \alpha^{2nd}(L)$, $m = \min(L_{=\alpha}) < M = \max(L_{=\alpha})$.

Example 5. Let $L = \langle 1/10, 0.5/20, 0.9/30, 1/40, 1/50\rangle$ and $L' = \langle 0.2/10, 0.9/20, 0.9/30, 1/40\rangle$ be two simple lotteries, then: $L_{=1} = \{10, 40, 50\}$, $L^{core} = L^{2nd} = \langle 1/10, 1/50\rangle$, $L'_{=1} = \{40\}$, $L'^{core} = \langle 1/40\rangle$ and $L'^{2nd} = \langle 0.9/20, 0.9/30, 1/40\rangle$.

When two lotteries L and L' are to be compared w.r.t. LN, the decision is made at possibility degree 1 (i.e. the core level) as soon as one of them (say L) is not unimodal (and thus L^{2nd} is of the form $\langle 1/m, 1/M\rangle$). Indeed,

(i) If L'^{2nd} is of the form $\langle 1/m', 1/M'\rangle$ then $L \succ_{LN} L'$ when $m \geqslant M'$, $L' \succ_{LN} L$ when $m' \geqslant M$ and $L \sim_{LN} L'$ otherwise.
(ii) If L'^{2nd} is of the form $\langle \alpha'/m', 1/u', \alpha'/M'\rangle$ then $L \succ_{LN} L'$ when $m \geqslant u'$, $L' \succ_{LN} L$ when $u' \geqslant M$ and $L \sim_{LN} L'$ otherwise.

Let us now study the case where L and L' are unimodal, then L^{2nd} (resp. L'^{2nd}) can be written as $\langle \alpha/m, 1/u, \alpha/M \rangle$ (resp. $\langle \alpha'/m', 1/u', \alpha'/M' \rangle$).

(i) If $u \neq u'$, then only the modes are used to make the decision: either $u > u'$, and then $L \succ_{LN} L'$; or $u' > u$ and then $L' \succ_{LN} L$.

(ii) If $u = u'$, decision is made at level $\max(\alpha, \alpha')$; when $\alpha > \alpha'$, then either $u > M$ (and then $L' \succ_{LN} L$); or $u < m$ (and then $L \succ_{LN} L'$); or $M > u > m$ (and then $L \sim_{LN} L'$). Similar results hold for $\alpha' > \alpha$. $u' = u$ and $\alpha' = \alpha$, it is easy to show that the two lotteries are indifferent w.r.t. LN.

We can now show that:

Proposition 8. *$\succeq_{LN}$ satisfies the weak monotonicity principle.*

Since when comparing L and L', only the information contained in L'^{2nd} and L^{2nd} matters we can also show that:

Proposition 9. *$\forall L, L' \in \mathcal{L}, L \succeq_{LN} L' \iff L^{2nd} \succeq_{LN} L'$ and $L' \succeq_{LN} L \iff L' \succeq_{LN} L^{2nd}$.*

This means that we can replace any simple lottery by its 2nd core lottery during the computation without changing the set of optimal strategies of the problem. A second consequence of that proposition is that if $L^{2nd} = L'^{2nd}$ then for any L'', $L \succeq_{LN} L'' \iff L' \succeq_{LN} L''$ and $L'' \succeq_{LN} L \iff L'' \succeq_{LN} L'$. Lotteries sharing the same 2nd core form an equivalence class. We shall then remove from S the pairs (L^{2nd}, δ) such that there exist in the set another pair (L'^{2nd}, δ') where $L^{2nd} = L'^{2nd}$: since leading to the same 2nd core lottery, δ and δ' are equivalent w.r.t. LN. The set S will thus contain pairs (L_i, δ_i) where each L_i is indifferent to others, while no pair of lotteries in S share the same 2nd core lottery. Now, it is possible to prove that:

Proposition 10. *Let L and L' be two indifferent 2nd core lotteries of the form $L = \langle 1/u, \alpha/m, \alpha/M \rangle$ and $L' = \langle 1/u', \alpha'/m', \alpha'/M' \rangle$ such that $\alpha < \alpha'$. Then:*

$$L \sim_{LN} L' \quad \textit{iff} \quad u = u' \textit{ and } m < u < M.$$

This means that in a set S of indifferent 2nd core lotteries, all the unimodal 2nd core lotteries share the same mode and for all of them, but maybe the one with the lowest α^{2nd}, and $m < u < M$. This latter is very significant, since one can forget about all the unimodal 2nd core lotteries in the set S for which α^{2nd} is not the lowest one. It can indeed be shown that:

Proposition 11. *Let L and L' be two indifferent 2nd core lotteries of the form $L = \langle 1/u, \alpha/m, \alpha/M \rangle$ and $L' = \langle 1/u, \alpha'/m', \alpha'/M' \rangle$ where $\alpha < \alpha'$. Let $L_1 = \langle \gamma/L, \beta/L'' \rangle$ and $L_2 = \langle \gamma/L', \beta/L'' \rangle$ such that L'' is a lottery and $\max(\gamma, \beta) = 1$. Then for any lottery L_b, it holds that:*

(i) $L_2 \succ_{LN} L_b \Rightarrow L_1 \succ_{LN} L_b$;

(ii) $L_b \succ_{LN} L_2 \Rightarrow L_b \succ_{LN} L_1$.

Actually, replacing L' by L in a strategy may break ties (for instance, it may happen that $\langle \gamma/L', \beta/L'' \rangle \sim_{LN} L_b$ and $\langle \gamma/L, \beta/L'' \rangle \succ_{LN} L_b$), but this can never change the strict preference. An important consequence of Proposition 11 is that, if the strategy corresponding to a lottery L' belongs to the set of optimal lotteries, so does the strategy associated to L.

Thus, before returning the set S of lotteries computed by the recursive procedure, the pruning function Prune_O (see Algorithm 3) can prune not only the dominated lotteries (and the associated strategies) but also all the unimodal core lotteries except the one with the lowest α^{2nd}. As a consequence, the sets of lotteries handled by Extended Dynamic Programming when optimizing LN contains at most $2.n^2$ elements (n^2 multimodal core lotteries and n^2 unimodal 2nd core lotteries sharing the same α^{2nd}), all being indifferent to each other. The Extended Dynamic Programming procedure is thus polynomial in the size of the decision tree when $O = LN$.

In summary, for both $L\Pi$ and LN, it is possible to ensure the polynomiality of the size of the set of strategies handled by Extended Dynamic Programming. Hence the result:

Proposition 12. *DT-OPT-LN and DT-OPT-LΠ belong to P.*

3.3.3. Order of Magnitude Expected Utility

In kappa decision trees, for any $C_i \in \mathcal{C}$ the uncertainty pertaining to the more or less possible outcomes N of C_i is represented by a kappa degree κ_i:

$$\forall N \in Succ(C_i), \quad \kappa_i(N) = Magnitude\big(P\big(N|path(C_i)\big)\big).$$

Algorithm 2: ExtProgDyn(N: node).

```
Result: A set $S$ of pairs $(L, \delta_L)$
Data: $O$ is the criterion to optimize ($O = L\Pi$ or $O = LN$).
Constant: The empty strategy $\delta_0$
begin
    if $N \in \mathcal{LN}$ // Leaves
    then
        $\forall u_i, L[u_i] \leftarrow 0$;
        $L[u(N)] \leftarrow 1$;
        $S \leftarrow \{(L, \delta_0)\}$;
    if $N \in \mathcal{C}$ // Chance nodes
    then
        // Reduce the compound lotteries
        foreach $Y_j \in Succ(N)$ do $S_j \leftarrow ExtProgDyn(Y_j)$   $S \leftarrow S_1$;
        for $j \in \{2, \ldots, |Succ(N)|\}$ do
            $Res \leftarrow \emptyset$;
            foreach $(L, \delta) \in S, (L', \delta') \in S_j$ do
                // Create a strategy $\delta''$ and a lottery $L''$
                $\delta'' \leftarrow \delta \cup \delta'$;
                $\forall u_i, L''[u_i] \leftarrow 0$;
                for $i \in \{1, \ldots, n\}$ do
                    $L''[u_i] \leftarrow \max(L[u_i], (\otimes(\pi_N(Y_j), L'[u_i])))$;
                    if $O = L\Pi$ then $L'' \leftarrow L''^{core}$
                    if $O = LN$ then $L'' \leftarrow L''^{2nd}$
                    $Res \leftarrow Res \cup \{(L'', \delta'')\}$;
            $S \leftarrow Prune_O(Res)$;
    if $N \in \mathcal{D}$ // Decision nodes
    then
        // Build the set of possible decisions
        $S \leftarrow \emptyset$;
        foreach $Y \in Succ(N)$ do  $S \leftarrow S \cup ExtProgDyn(Y)$ foreach $(L, \delta) \in S$ do  $\delta(N) \leftarrow L$ $S \leftarrow Prune_O(S)$;
    return $S$;
```

Algorithm 3: Prune$_O$(S: a set S of pairs (L: strategy, δ_L: lottery)).

```
Data: $O$ is the criterion to optimize ($O = L\Pi$ or $O = LN$)
Result: A pruned set $Res$ of pairs $(L', \delta'_L)$
begin
    $Res \leftarrow S$;
    foreach $(L, \delta) \in S$ do
        foreach $(L', \delta') \in S, \delta' \neq \delta$ do
            if $(L' \succ_O L)$ or $(L$ is identical to $L')$ then
                Remove $(L, \delta)$ from $Res$;
        if $O = LN$ then
            if $(L = \langle 1/u, \alpha/m, \alpha/M \rangle)$ and $(\exists L' = \langle 1/u, \alpha'/m', \alpha'/M' \rangle \in S)$, with $\alpha' < \alpha$ then
                Remove $(L, \delta)$ from $Res$;
    return $Res$;
```

Such that the normalization condition that the degree $\kappa = 0$ is given to at least one N in $Succ(C_i)$. According to the interpretation of kappa-ranking in terms of order of magnitude of probabilities, the product of infinitesimal conditional probabilities along the paths lead to a sum of the kappa levels. Hence the following principle of reduction of the kappa lotteries:

$$Reduction\bigl(\langle \kappa_1/L_1, \ldots, \kappa_k/L_k \rangle\bigr) = \Bigl\langle \min_{j=1..k}\bigl(\kappa_1^j + \kappa_j\bigr)/u_1, \ldots, \min_{j=1..k}\bigl(\kappa_m^j + \kappa_j\bigr)/u_m \Bigr\rangle. \tag{22}$$

It follows from the definition of the order of magnitude expected utility (i.e. $OMEU(L = \langle \kappa_1/\mu_1, \ldots, \kappa_n/\mu_n \rangle) = \min_{i=1..n}\{\kappa_i + \mu_i\}$) that this criterion is transitive and also satisfies the weak monotonicity principle:

Proposition 13. *$\succeq_{OMEU}$ is complete, transitive and satisfies the weak monotonicity property.*

As a consequence Dynamic Programming is sound and complete for the optimization of Order of Magnitude Expected Utility:

Corollary 2. *DT-OPT-OMEU belongs to P.*

3.3.4. Possibilistic Choquet integrals (Ch_N, Ch_Π)

Contrary to qualitative utilities, binary possibilistic utility and likely dominance, the situation is much lesser comfortable for Choquet integrals (either Ch_N or Ch_Π). The point is that the possibilistic Choquet integrals (as many other Choquet integrals) do not satisfy the monotonicity principle neither in the ordinal setting nor in the numerical one. This is illustrated by the following counterexamples.

Counterexample 1 *(Ch_N does not satisfy the monotonicity principle).* Let $L = \langle 0.2/0, 0.5/0.51, 1/1\rangle$, $L' = \langle 0.1/0, 0.6/0.5, 1/1\rangle$, $L'' = \langle 0.01/0, 1/1\rangle$, $L_1 = \langle \alpha/L, \beta/L''\rangle$ and $L_2 = \langle \alpha/L', \beta/L''\rangle$, with $\alpha = 0.55$ and $\beta = 1$.

- For $\otimes = \min$, we have: $Reduction(L_1) = \langle 0.2/0, 0.5/0.51, 1/1\rangle$ and $Reduction(L_2) = \langle 0.1/0, 0.55/0.5, 1/1\rangle$.
Computing $Ch_N(L) = 0.653$ and $Ch_N(L') = 0.650$ we get $L \succcurlyeq_{Ch_N} L'$.
But $Ch_N(Reduction(L_1)) = 0.653 < Ch_N(Reduction(L_2)) = 0.675$, i.e. $\langle \alpha/L, \beta/L''\rangle \prec_{Ch_N} \langle \alpha/L', \beta/L''\rangle$, which contradicts the monotonicity property.
- For $\otimes = *$, we have: $Reduction(L_1) = \langle 0.11/0, 0.275/0.51, 1/1\rangle$ and $Reduction(L_2) = \langle 0.055/0, 0.33/0.5, 1/1\rangle$.
Computing $Ch_N(L) = 0.653$ and $Ch_N(L') = 0.650$ we get $L \succcurlyeq_{Ch_N} L'$.
But $Ch_N(Reduction(L_1)) = 0.809 < Ch_N(Reduction(L_2)) = 0.45$, i.e. $\langle \alpha/L, \beta/L''\rangle \prec_{Ch_N} \langle \alpha/L', \beta/L''\rangle$, which contradicts the monotonicity property.

Counterexample 2 *(Ch_Π does not satisfy the monotonicity principle).* Let $L = \langle 1/0, 0.5/0.51, 0.2/1\rangle$, $L' = \langle 1/0, 0.6/0.5, 0.1/1\rangle$ and $L'' = \langle 1/0, 0.49/0.51\rangle$. $L_1 = \langle \alpha/L, \beta/L''\rangle$ and $L_2 = \langle \alpha/L', \beta/L''\rangle$, with $\alpha = 1$ and $\beta = 0.55$.

- For $\otimes = \min$, we have: $Reduction(L_1) = \langle 1/0, 0.5/0.51, 0.2/1\rangle$ and $Reduction(L_2) = \langle 1/0, 0.6/0.5, 0.49/0.51, 0.1/1\rangle$.
Computing $Ch_\Pi(L) = 0.353$ and $Ch_\Pi(L') = 0.350$ we get $L \succ_{Ch_\Pi} L'$.
But $Ch_\Pi(Reduction(L_1)) = 0.3530 \prec Ch_\Pi(Reduction(L_2)) = 0.3539$, i.e. $\langle \alpha/L, \beta/L''\rangle \prec_{Ch_\Pi} \langle \alpha/L', \beta/L''\rangle$, which contradicts the monotonicity property.
- For $\otimes = *$, we have: $Reduction(L_1) = \langle 1/0, 0.5/0.51, 0.2/1\rangle$ and $Reduction(L_2) = \langle 1/0, 0.6/0.5, 0.26/0.51, 0.1/1\rangle$.
Computing $Ch_\Pi(L) = 0.353$ and $Ch_\Pi(L') = 0.350$ we get $L \succ_{Ch_\Pi} L'$.
But $Ch_\Pi(Reduction(L_1)) = 0.3530 < Ch_\Pi(Reduction(L_2)) = 0.3516$, i.e. $\langle \alpha/L, \beta/L''\rangle \prec_{Ch_\Pi} \langle \alpha/L', \beta/L''\rangle$, which contradicts the monotonicity property.

As a consequence, Dynamic Programming cannot guarantee an optimal policy. Making a step further, we show that it is impossible to solve the problem in polytime, unless $P = NP$:

Proposition 14. *DT-OPT-Ch_N and DT-OPT-Ch_Π are NP-complete.*

The proofs of NP-hardness of these criteria (detailed in Appendix A) follow the principle used by [16] for proving the difficulty of the problem in the RDU case. For DT-OPT-Ch_Π, the transformation we used is a simplification of the one proposed in [16]. For DT-OPT-Ch_N, the underlying idea is different and is mainly based on the fact that compounding a lottery with another lottery that is not strictly better than the original one cannot increase its Choquet value (see Section 2.6).

In summary, optimizing possibilistic Choquet integrals is untractable in the general case. We can nevertheless look restrictions that make the question polynomial. It is for instance the case when the decision tree is "binary", i.e. when the leaves involve two levels of utility.

Proposition 15. *DT-OPT-Ch_N (resp. DT-OPT-Ch_Π) is polynomial on decision trees involving two levels of utility.*

4. A Branch and Bound approach for optimizing the Choquet-based possibilistic criteria

As we have seen in the previous section, possibilistic Choquet integrals define NP-hard problems and the polynomial algorithm of Dynamic Programming may lead to sub-optimal decision policies. As an alternative, we have chosen to proceed by implicit enumeration via a Branch and Bound algorithm, following [15] for the case of another (non-possibilistic) Choquet integrals, namely the one encoding the Rank Dependent Utility criterion. The Branch and Bound procedure (see Algorithm 4) takes as argument a partial strategy δ and an upper bound of the Choquet value Ch_μ of the best extension of the partial strategy (μ is either the necessity or the possibility measure, depending on the criterion that is to be optimized). It returns the Choquet value of the best strategy found so far, δ^{opt}. As initial value for δ we retain the empty strategy ($\delta(D_i) = \bot, \forall D_i$). For δ^{opt}, we can choose the strategy provided by the Dynamic Programming algorithm: indeed, even not necessarly providing an optimal strategy, this algorithm generally provides a good one. At each step, the current partial strategy, δ, is developed by the choice of an action for some unassigned decision node. When several decision nodes are candidate, the one with the minimal rank (i.e. the former one according to the temporal order) is developed first. The recursive procedure backtracks when either the current strategy is complete (then δ^{opt} and Ch_μ^{opt} may be updated) or proves to be worst than the current δ^{opt} in any case.

Algorithm 4: BB(δ: a (partial) strategy, Ch_μ: its Choquet value) with $\mu = N$ or $\mu = \Pi$.

```
Result: $Ch_\mu^{opt}$ i.e. the value of $\delta^{opt}$, the best strategy found so far
begin
    if $\delta = \emptyset$ then $\mathcal{D}_{pend} \leftarrow \{D_0\}$ else
        $D_{pend} \leftarrow \{D_i \in \mathcal{D}$ s.t. $\delta(D_i) = \perp$ and $\exists D_j, \delta(D_j) \neq \perp$ and $D_i \in Succ(\delta(D_j))\}$;
    // Is $\delta$ a complete strategy?
    if $\mathcal{D}_{pend} = \emptyset$ then
        if $Ch_\mu^\delta > Ch_\mu^{opt}$ then
            $\delta^{opt} \leftarrow \delta$;
            $Ch_\mu^{opt} \leftarrow Ch_\mu^\delta$;
    else
        $D_{next} \leftarrow \arg\min_{D_i \in D_{pend}} i$;
        foreach $C_i \in Succ(D_{next})$ do
            $\delta(D_{next}) \leftarrow C_i$;
            $Eval \leftarrow Ch_\mu(Lottery_\mu(D_0, \delta))$;
            if $Eval > Ch_\mu^{opt}$ then $Ch_\mu^{opt} \leftarrow BranchAndBound(\delta, Ch_\mu^\delta)$
    return $Ch_\mu^{opt}$;
```

4.1. Computation of an upper bound for the current partial solution

In order to get an upper bound of the Choquet value of the best completion of δ, we call a function $Lottery_\mu$ ($\mu = N$ or Π) that computes a lottery that is better than all the complete strategies that extend δ when applied from a node given in argument and use the Choquet value of this lottery as an upper bound of the Choquet value of the best strategy compatible with δ. Whenever $Ch_\mu(Lottery_\mu(\delta)) \leqslant Ch_\mu^{opt}$ the algorithm will backtrack, yielding the choice of another action for the last decision nodes considered. Moreover when δ is complete, $Lottery_\mu(\delta)$ returns $L(D_0, \delta)$; the upper bound is equal to the Choquet value when computed for a complete strategy.

$Lottery_\mu(\delta)$ actually builds a lottery that *overcomes* with respect of μ all the possible extensions of δ. Formally:

Definition 2. A lottery $L \in \mathcal{L}$ is said to overcome a lottery $L' \in \mathcal{L}$ w.r.t. N (or "to possibility overcome L'") iff: $\forall u_i, N(L \geqslant u_i) \geqslant N(L' \geqslant u_i)$.

A lottery $L \in \mathcal{L}$ is said to overcome a lottery $L' \in \mathcal{L}$ w.r.t. Π ("to necessarily overcome L'") iff: $\forall u_i, \Pi(L \geqslant u_i) \geqslant \Pi(L' \geqslant u_i)$.

$Ch_N(Lottery_N(\delta))$ (resp. $Ch_\Pi(Lottery_\Pi(\delta))$) is an upper bound of the Choquet value of the best strategy compatible with δ. Indeed:

Proposition 16. *If L' necessarily (resp. possibly) overcomes L'', then $Ch_N(L') \geqslant Ch_N(L'')$ (resp. $Ch_\Pi(L') \geqslant Ch_\Pi(L'')$).*

Let us build an upper bound of the possibility-based Choquet integral. The computation relies on a $Lottery_\Pi$ (see Algorithm 5 bellow) that computes a lottery that overcomes all the possible extensions of the current strategy. This function inputs a partial strategy and proceeds backwards, assigning a simple lottery $\langle 1/u(L_i)\rangle$ to each terminal node LN_i. At each chance node C_i, it performs the composition of the lotteries in $Succ(C_i)$ according to Eq. (19). At each decision node D_i it builds a lottery that overcomes all those in $Succ(D_i)$ w.r.t. Π; this lottery is simply the upper envelop of all the lotteries in $Succ(D_i)$, i.e.

$$\forall i, \quad L_X = L^{upp}_{Succ(D_i)}(u_i) = \max_{L \in Succ(D_i)} L[u_i].$$

It follows directly that $L_X[u_i] \geqslant L[u_i]$ for any $L \in Succ(D_i)$, i.e. L_X overcomes each of the lotteries in $Succ(D_i)$. Hence, the Choquet value of $Lottery_\Pi(X, \delta)$ is an upper bound of the Choquet value of the best complete strategy rooted in X and extending δ, which proves the correctness of our algorithm for $\mu = \Pi$.

Let us now build an upper bound for the necessity-based Choquet integral. The computation relies on a function $Lottery_N$ (see Algorithm 6) which, as $Lottery_\Pi$, inputs a partial strategy and proceeds backwards, assigning a simple lottery $\langle 1/u(NL_i)\rangle$ to each terminal node LN_i. At each chance node C_i, it performs the composition of the lotteries in $Succ(C_i)$ according to Eq. (19). At each decision node D_i it builds a lottery that overcomes all those in $Succ(D_i)$ w.r.t. N. But the case of the necessity-based Choquet integrals is more tricky than the case of the possibility-based integrals. Let us use the following notations and definitions:

Algorithm 5: *Lottery*$_\Pi$: Computation of a lottery that overcomes w.r.t. Π all the completions of δ when applied from X.

```
Data: A node $X$, a (possibly partial) strategy $\delta$
Result: A lottery $L_X$
// $L_X[u_i]$ is the possibility degree to have the utility $u_i$ in $L_X$
begin
    foreach $i \in \{1,\dots,n\}$ do $L_X[u_i] \leftarrow 0$;                      // Initialization of $L_X$
    if $X \in \mathcal{LN}$ then $L_X[u(X)] \leftarrow 1$
    if $X \in \mathcal{C}$ then
        foreach $Y \in Succ(X)$ do
            $L_Y \leftarrow Lottery_\Pi(Y,\delta)$;
            foreach $i \in \{1,\dots,n\}$ do
                $L_X[u_i] \leftarrow \max(L_X[u_i], \otimes(\pi_X(Y), L_Y[u_i]))$;
            // $\otimes = \min$ in the ordinal setting; $\otimes = *$ in the numerical setting
    if $X \in \mathcal{D}$ then
        if $\delta(X) = \perp$ then
            foreach $Y \in Succ(X)$ do
                $L_Y \leftarrow Lottery_\Pi(Y,\delta)$;
                foreach $i \in \{1,\dots,n\}$ do
                    $L_X[u_i] \leftarrow \max(L_X[u_i], L_Y[u_i])$;
        else
            $L_X \leftarrow Lottery_\Pi(\delta(X),\delta)$;
    return $L_X$;
```

- Given a simple lottery $L \in \mathcal{L}$, G_L^c is the *possibilistic decumulative* function of L: $\forall u \in U, G_L^c(u) = N(L \geqslant u)$.
- Given a set $\mathcal{G} = \{G_{L_1}^c, \dots, G_{L_k}^c\}$ of decumulative functions, the *upper decumulative envelop* of $\mathcal{G}$ is the decumulative function $G_\mathcal{G}^c$ defined by:

$$\forall u \in U, \quad G_\mathcal{G}^c(u) = \max_{G_{L_i}^c \in \mathcal{G}} G_{L_i}^c(u).$$

- Given a decumulative function G^c on U, $Rev(G^c)$ is the lottery defined by:

$$Rev\big(G^c\big)(u_i) = \begin{cases} 1 & \text{if } i = n, \\ 1 - G^c(u_{i+1}) & \text{if } i \in \{1,\dots,n-1\}. \end{cases}$$

Now it is easy to show the following lemma:

Lemma 1. *The possibilistic decumulative function associated to a lottery $Rev(G^c)$ is equal to G^c.*

As a consequence of Lemma 1, we have:

Proposition 17. *Given a set $\{L_1,\dots,L_k\} \subseteq \mathcal{L}$ of simple lotteries over U and $\mathcal{G} = \{G_{L_1}^c,\dots,G_{L_k}^c\}$ the set of their decumulative functions. We have: $Rev(G_\mathcal{G}^c)$ overcomes any lottery $L_i \in \{L_1,\dots,L_k\}$.*

Hence, the Choquet value Ch_N of $Lottery_N(D_0,\delta)$ is an upper bound of the Choquet value of the best complete strategy compatible with δ, which proves the correctness of our algorithm for $\mu = N$.

4.2. Experimental results

The algorithms presented in the previous section have been implemented over Java and the computational experiments were carried out on a processor Intel Xeon X5650, 2.66 GHz, 16 GB of RAM. We have compared the performances of the Branch and Bound approach with those of the sole Dynamic Programming procedure. The tests were performed on complete binary decision trees with different height. We have considered four sets of problems, the number of decisions to be made in sequence (denoted *seq*) varies from 2 to 5, with an alternation of decision and chance nodes: At each decision level i (i.e. odd levels from 1 to $seq*2$), the tree contains 2^{i-1} decision nodes followed by 2^i chance nodes. Thus, the number of chance nodes in the tree is equal to $|\mathcal{C}| = |\mathcal{D}| * 2$ and the number of leaves is equal to $|\mathcal{LN}| = |\mathcal{D}| + |\mathcal{C}| + 1$. This means that for the set of problems with a sequence length $seq = 2$ (resp. 3, 4, 5), the number of decision nodes is equal to $|\mathcal{D}| = 5$ (resp. 21, 85, 341), i.e. the number of nodes in the tree is equal to $|\mathcal{N}| = |\mathcal{D}| + |\mathcal{C}| + |\mathcal{LN}| = 31$ (resp. 127, 511, 2047) nodes. (See Fig. 3.)

Algorithm 6: $Lottery_N$: Computation of a lottery that overcomes w.r.t. N all the completions of δ when applied from X.

```
Data: A node $X$, a (possibly partial) strategy $\delta$
Result: A lottery $L_X$
// $L_X[u_i]$ is the possibility degree to have the utility $u_i$ in $L_X$
begin
    foreach $i \in \{1, \dots, n\}$ do $L_X[u_i] \leftarrow 0$
    if $X \in \mathcal{LN}$ then $L_X[u(X)] \leftarrow 1$
    if $X \in \mathcal{C}$ then
        foreach $Y \in Succ(X)$ do
            $L_Y \leftarrow Lottery_N(Y, \delta)$;
            for $i \in \{1, \dots, n\}$ do
            $L_X[u_i] \leftarrow \max(L_X[u_i], \otimes(\pi_X(Y), L_Y[u_i]))$
            // $\otimes = \min$ in the ordinal setting; $\otimes = *$ in the numerical setting
    if $X \in \mathcal{D}$ then
        if $\delta(X) \neq \perp$ then
            $L_X \leftarrow Lottery_N(\delta(X), \delta)$;
        else
            if $|Succ(X)| = 1$ then
                $L_X \leftarrow Lottery_N(\delta(Succ(X)), \delta)$;
            else
                foreach $Y \in Succ(X) \cap N_\delta$ do
                    $L_Y \leftarrow Lottery_N(Y, \delta)$;
                    foreach $i \in \{1, \dots, n\}$ do
                        $G^c_Y[u_i] \leftarrow 1 - \max_{u_j < u_i} L_Y[u_j]$;
                // Compute the upper envelop of $G^c$
                foreach $i \in \{1, \dots, n\}$ do
                    $G^c[u_i] \leftarrow \max_{Y \in Succ(X) \cap N_\delta} G^c_Y[u_i]$;
                // Compute $Rev(G^c)$
                $L_X[u_n] \leftarrow 1$;
                foreach $i \in \{n-1, \dots, 1\}$ do
                    $L_X[u_i] \leftarrow 1 - G^c[u_{i+1}]$;
    return $L_X$;
```

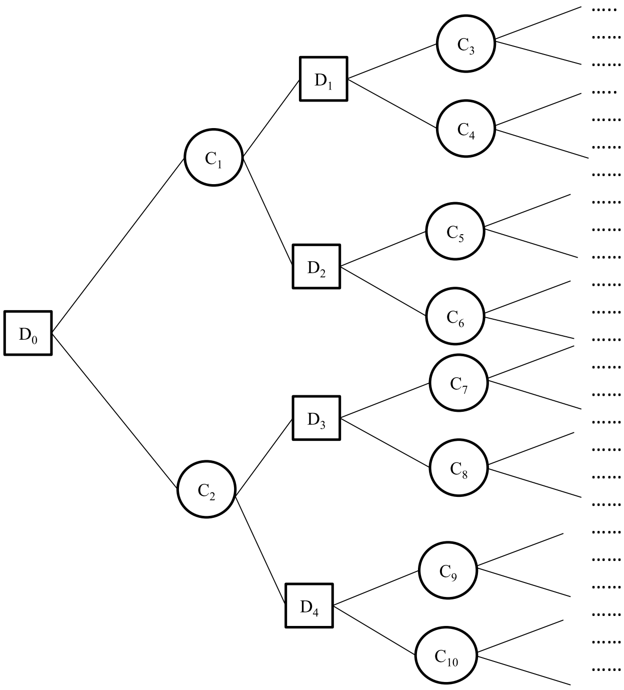

Fig. 3. Structure of the randomly generated decision trees for $seq = 2$.

The values of the utilities have been randomly fired in the set $U = \{0, 1, \dots, 20\}$ following equiprobable distribution. Conditional possibilities relative to chance nodes are also chosen randomly in $[0, 1]$ and normalized. Each of the four samples of problems contains 300 randomly generated problems.

Table 1
Percentage of problems for which the DP value is correct.

		Length *seq* of the sequence			
		2	3	4	5
Ch_N	Qualitative	91%	76%	51%	34%
	Numerical	96%	86%	62%	41%
Ch_Π	Qualitative	95%	86%	92%	99%
	Numerical	95%	88%	93%	98%

Table 2
Average closeness value $\frac{V_{DP}}{V_{BB}}$ for problems in which $V_{DP} \neq V_{BB}$.

		Length *seq* of the sequence			
		2	3	4	5
Ch_N	Qualitative	95.0%	95.6%	94.7%	94.7%
	Numerical	94%	97%	97%	98%
Ch_Π	Qualitative	96.4%	97.8%	98.5%	99.1%
	Numerical	97%	98%	98%	99%

Table 3
Execution CPU time for Ch_N (in seconds).

		Length *seq* of the sequence			
		2	3	4	5
Qualitative setting	ProgDyn	0.013	0.014	0.083	0.30
	BB	0.05	0.1	3.77	55 214
Numerical setting	ProgDyn	0.02	0.02	0.06	0.275
	BB	0.05	0.07	3.5	42 362

Table 4
Execution CPU time for Ch_Π (in seconds).

		Length *seq* of the sequence			
		2	3	4	5
Qualitative setting	ProgDyn	0.01	0.04	0.07	0.15
	BB	0.023	0.103	1.943	4817.753
Numerical setting	ProgDyn	0.01	0.026	0.033	0.235
	BB	1	1.07	1.426	395.13

Quality of solutions provided by Dynamic Programming Since the application of Dynamic Programming can lead to sub-optimal strategies, we propose to estimate their quality by comparing them to exact values generated by Branch and Bound. More precisely, we compute for different trees the number of cases for which the value provided by Dynamic Programming is not the optimal one, and for the problems on which it fails, we report the closeness value equal to $\frac{V_{DP}}{V_{BB}}$ where V_{DP} is the possibilistic Choquet integrals relative to the optimal strategy provided by Dynamic Programming and V_{BB} by Branch and Bound – the results are summarized in Tables 1 and 2.

For the necessity-based qualitative Choquet integrals the percentage of full success of the DP approximation is decreasing with the number of decision to be made. Notice that this approximation is very good even when it does not predict the optimal value, with a closeness value greater than 94% in any case. For the possibility-based qualitative Choquet integrals, the closeness value is also very good (greater than 96% in any case) and the percentage of full success of the DP approximation is increasing with the number of decision to be made (this is due to the fact that the higher the number of leaves, the higher the possibility of getting a Ch_Π value maximal in the utility scale – to utility degree 20 in our case; for trees with 2047 nodes indeed, all the problems have a Ch_Π equal to 20).

The results are similar when the conditioning is numerical (product-based) rather than qualitative (min-based).

Tables 3 and 4 provide different average execution CPU time for Dynamic Programming and for the complete Branch and Bound procedure.

Obviously, the execution CPU time for Ch_N and also for Ch_Π in qualitative and numerical setting increases according to the size of the tree, and is better for Dynamic Programming than for the complete than Branch and Bound algorithm. Unsurprisingly, the increasing is linear in the first case and clearly exponential for the latter one. But the times remains

Table 5
Results about the complexity of DT-OPT-O.

U_{pes}	U_{opt}	PU	$L\Pi$	LN	OMEU	Ch_N	Ch_Π
P	P	P	P	P	P	NP-complete	NP-complete

affordable in average even for very big trees (341 nodes). For reasonable trees (85 decision nodes) the average CPU time is less than 4 s and the maximal CPU time is about one minute. The results are thus good enough to allow the handling of real-size problems.

5. Conclusion

In this paper, we have shown that strategy optimization in possibilistic decision trees is tractable for most of the criteria, extending the results about the qualitative utility criteria U_{pes} and U_{opt} to other possibilistic criteria. We have in particular shown that even not fully transitive, the possibilistic likely dominance criteria (LN, L_Π) can be handled in polytime: the number of strategies retained in the set-Extended algorithm of Dynamic Programming can be kept polynomial throughout the process. We have also shown that the problem is intractable for the possibilistic Choquet-based criteria. Finally, we have extended this work to OMEU, defining a new model for sequential decision trees, extending the notion of reduction to kappa lotteries and showing that this model obeys the weak monotonicity principle. These results are summarized in Table 5.

It should be noticed that the optimization of the possibilistic Choquet integrals is "only" in NP: the computation of the Choquet value of a possibilistic strategy is polynomial, whereas this computation can be more costly for other capacity measures; for instance computing the Choquet value of a strategy on the basis of its multi prior expected utility is itself an NP-hard problem – and the corresponding optimization problem is probably beyond NP. Concerning possibilistic Choquet integrals, we have then proposed a Branch and Bound approach that extends the use of Dynamic Programming. Our first experiments suggest that this approach is computationally sustainable. So, it appears that the use of possibilistic decision criteria does not lead to an increase in complexity, except for Choquet integrals. This is an interesting result that allows the extension of our work to possibilistic influence diagrams. Further work includes the development of a direct evaluation algorithm for possibilistic influence diagrams where possibilistic Choquet integrals are used as a decision criteria inspired by the variable elimination approach [16].

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improving.

Appendix A

Proof of Proposition 1. Let u_k be the best possible utility in L, i.e. $u_k = \max_{u_j \in L, \lambda_j > 0} u_j$, $\lambda_k \leqslant 1$ (i.e. is not necessarily equal to 1), but $\exists j, 1 \leqslant j \leqslant k$ s.t. $\lambda_j = 1$ and thus $\forall j, k \leqslant j \leqslant n, 1 - \max(\lambda_1, \dots, \lambda_j) = 0$.

L and L' are written as follows:

$$L = \langle \lambda_1/u_1, \dots, \lambda_{i-1}/u_{i-1}, \lambda_i/u_i, \lambda_{i+1}/u_{i+1}, \dots, \lambda_{k-1}/u_{k-1}, \lambda_k/u_k, \lambda_{k+1}/u_{k+1}, \dots, \lambda_n/u_n \rangle,$$
$$L' = \langle \lambda_1/u_1, \dots, \lambda_{i-1}/u_{i-1}, \lambda'_i/u_i, \lambda_{i+1}/u_{i+1}, \dots, \lambda_{k-1}/u_{k-1}, \lambda_k/u_k, \lambda_{k+1}/u_{k+1}, \dots, \lambda_n/u_n \rangle.$$

This means that $Ch_N(L)$ can be decomposed into three terms, i.e. $Ch_N(L) = V_1 + V_2 + V_3$ where:

$$\begin{aligned}
V_1 &= u_1 + (u_2 - u_1)(1 - \lambda_1) + \dots + (u_i - u_{i-1})\big(1 - \max(\lambda_1, \dots, \lambda_{i-1})\big),\\
V_2 &= (u_{i+1} - u_i)\big(1 - \max(\lambda_1, \dots, \lambda_i)\big)\\
&\quad + (u_{i+2} - u_{i+1})\big(1 - \max(\lambda_1, \dots, \lambda_i, \lambda_{i+1})\big) + \cdots\\
&\quad + (u_k - u_{k-1})\big(1 - \max(\lambda_1, \dots, \lambda_i, \lambda_{i+1}, \dots, \lambda_{k-1})\big),\\
V_3 &= (u_{k+1} - u_k)\big(1 - \max(\lambda_1, \dots, \lambda_k)\big) + \cdots\\
&\quad + (u_n - u_{m-1})\big(1 - \max(\lambda_1, \dots, \lambda_n)\big)\\
&= 0.
\end{aligned}$$

(Since $\forall j, k \leqslant j \leqslant n, 1 - \max(\lambda_1, \dots, \lambda_j) = 0$.)

Thus $Ch_N(L) = V_1 + V_2$.

In the same manner $Ch_N(L')$ can be decomposed into three terms, i.e. $Ch_N(L') = V'_1 + V'_2 + V'_3$ where:

$$\begin{aligned}
V'_1 &= u_1 + (u_2 - u_1)(1 - \lambda_1) + \cdots + (u_i - u_{i-1})\big(1 - \max(\lambda_1, \ldots, \lambda_{i-1})\big)\\
&= V_1,\\
V'_2 &= (u_{i+1} - u_i)\big(1 - \max(\lambda_1, \ldots, \lambda'_i)\big)\\
&\quad + (u_{i+2} - u_{i+1})\big(1 - \max(\lambda_1, \ldots, \lambda'_i, \lambda_{i+1})\big) + \cdots\\
&\quad + (u_k - u_{k-1})\big(1 - \max(\lambda_1, \ldots, \lambda'_i, \lambda_{i+1}, \ldots, \lambda_{k-1})\big),\\
V'_3 &= (u_{k+1} - u_k)\big(1 - \max(\lambda_1, \ldots, \lambda_k)\big) + \cdots\\
&\quad + (u_n - u_{m-1})\big(1 - \max(\lambda_1, \ldots, \lambda_n)\big)\\
&= V_3 = 0.
\end{aligned}$$

As a consequence, it holds that: $Ch_N(L) - Ch_N(L') = V_2 - V'_2$.

Since $\lambda'_i > \lambda_i$, thus $1 - \max(\lambda_1, \ldots, \lambda'_i) \leqslant 1 - \max(\lambda_1, \ldots, \lambda_i)$ and $\forall j, i < j < k$, $1 - \max(\lambda_1, \ldots, \lambda'_i, \ldots, \lambda_j) \leqslant 1 - \max(\lambda_1, \ldots, \lambda_i, \ldots, \lambda_j)$.

Thus $V'_2 \leqslant V_2$ and $Ch_N(L') \leqslant Ch_N(L)$. □

Proof of Proposition 2. Let $L = Reduction(\langle 1/L_1, 1/L_2\rangle)$.

Since $\forall \lambda, (\lambda * 1) = \lambda$ and $\min(1, \lambda) = \lambda$, Eq. (5) gives the same L for both $\otimes = *$ and $\otimes = \min$, i.e. $\forall j \in \{1..m\}, \lambda_j = \max(\lambda^1_j, \lambda^2_j)$.

Since $\max_{u_i \in L_2, \lambda^2_i > 0} u_i \leqslant \max_{u_i \in L_1, \lambda^1_i > 0} u_i$, we can get L from L_1 in n steps by replacing at each step $j \in \{1..m\}$, λ^1_j by $\max(\lambda^1_j, \lambda^2_j)$. According to Proposition 1, this is done without increasing the value of the Choquet integral of L. Formally, let $L_0 = L_1$, then for $j = 1..n$, $L_j = \langle \lambda^j_1/u_1, \ldots, \lambda^j_n/u_n\rangle$ such that for any $k \neq j$, $\lambda^j_k = \lambda^{j-1}_k$ and $\lambda^j_j = \max(\lambda^{j-1}_j, \lambda^{j-1}_2)$. By construction, $L_n = L$. Thanks to Proposition 1, $\forall j \in \{1..m\}, Ch_N(L_j) \leqslant Ch_N(L_{j-1})$.

Then $Ch_N(L_n) \leqslant Ch_N(L_0)$, i.e. $Ch_N(L) \leqslant Ch_N(L_1)$. □

Proof of Proposition 3. Let $\delta \in \Delta$ such that $\delta = \{(D_0, \delta(D_0)), \ldots, (D_i, \delta(D_i)), \ldots, (D_n, \delta(D_n))\}$ be a complete and sound strategy. We first compute the compound lottery corresponding to δ, merging each decision node D_i in δ with the chance node in $\delta(D_i)$, say C^δ_i. We get a compound lottery $L = \{C^\delta_0, \ldots, C^\delta_i, \ldots, C^\delta_l\}$; the merging is performed linearly in the number of decision nodes in the strategy.

Then we can suppose without loss of generality that the nodes are numbered in such a way that $i < j$ implies that C^δ_i does not belong to the subtree rooted by C^δ_j (we label the nodes from the root to the leaves). Then, for $i = m$ to 1, we replace each compound lottery $C^\delta_i = \langle \pi_i(X_{i1})/X_{i1}, \ldots, \pi_i(X_{ik_i})/X_{ik_i}\rangle$ by its reduction.

Since we proceed from the leaves to the root, the X_{i1} $(i = 1..m)$ are simple lotteries. Since the min, $*$ and max operators can be performed in linear time and size, the reduction of this 2 level compound lottery is linear in the size of the compound lottery. The size of the resulting compound lottery is bounded by the sum of the size of the elementary lotteries before reduction, and thus linear. In any case, it is bounded by the number of levels in the scale, which is itself bounded by the number of edges and leaves in the tree (for the case where all the possibility degrees and all the utility degrees are different). Hence a complexity of the reduction is bounded by $O(|E + LN|)$, where E is the number of edges and LN is the number of leave nodes in the strategy.

Thank to the backward recursion, each node in the strategy is visited only once. Thus a global complexity is bounded by $O(C * (E + LN))$, where E is the number of edges, LN the number of leave nodes and C the number of chance nodes in the strategy. □

Proof of Proposition 4. The principle of the Backward induction method at work in Dynamic Programming (Algorithm 1) is to eliminate sub-strategies that are not better than the optimal sub-strategies. The weak monotonicity principle expressed by Eq. (21) guarantees that the elimination of sub-strategies that are not strictly better than their concurrents is sound and complete for the decision trees of size 2. Notice that $L \succ_O L'$ does not imply that L' does not belong to an optimal strategy[3]; but it implies that if L' belongs to an optimal strategy, so does L. When a unique strategy among the optimal ones is searched for, the algorithm can forget about L'.

[3] That is why we called the property "weak monotonicity" – strong monotonicity is expressed by: $L \succcurlyeq_O L'$ iff $\langle \alpha/L, \beta/L''\rangle \succcurlyeq_O \langle \alpha/L', \beta/L''\rangle$. Strong monotonicity thus adds to weak monotonicity that if $L \succ_O L'$ then L' does not belong to an optimal strategy.

The sequel on the proof is by recursion on the depth of the decision tree. The property obviously holds for decision trees of depth 1:

- If the first node is a chance node, no decision is to be made and there is only one strategy, the empty one;
- If the first node is a decision one, the transitivity of $\succcurlyeq_O$ guarantees that the *max* value obtained at Line 2 (Algorithm 1) is not strictly lower (according to $\succ_O$) than the others.

Suppose that the property holds for decision trees of depth n. Consider a tree of depth $n+1$:

- If the first node is a chance node, the monotonicity principle ensures that the union of the sub-strategies given by the recursive calls is not dominated according to $\succ_O$;
- If the first node (say N) is a decision node, the procedure is called recursively on each of its children Y_i. Let $\Delta(Y_i)$ be the set of possible strategies for the sub-decision trees rooted in Y_i; the recursive call on each Y_i returns a strategy δ_i^* such that $\forall \delta' \in \Delta(Y_i)$, $\delta_i^* \succcurlyeq_O \delta_i'$ (by hypothesis of recursion); then by transitivity of $\succcurlyeq_O$, the strategy $\delta^* \succcurlyeq_O \delta_i^*$, $\forall Y_i \in Succ(N)$ is such that $\delta^* \succcurlyeq_O \delta'$, $\forall \delta' \in \bigcup_i \Delta(Y_i)$. □

Proof of Proposition 5. – *Weak monotonicity of PU*

Consider any three lotteries L, L' and L''. We can suppose without loss of generality that they are in a reduced form, i.e.:

$$L = \langle \bar{u}/\top, \underline{u}/\bot \rangle, \qquad L' = \langle \bar{v}/\top, \underline{v}/\bot \rangle \quad \text{and} \quad L'' = \langle \overline{w}/\top, \underline{w}/\bot \rangle.$$

Let

$$L_1 = Reduction\big(\langle \alpha/L, \beta/L'' \rangle\big) \quad \text{and} \quad L_2 = Reduction\big(\langle \alpha/L', \beta/L'' \rangle\big).$$

According to the reduction operation, we get:

$$L_1 = \langle \overline{u_1}/\top, \underline{u_1}/\bot \rangle \quad \text{and} \quad L_2 = \langle \overline{u_2}/\top, \underline{u_2}/\bot \rangle$$

where $\overline{u_1} = \max(\otimes(\alpha, \bar{u}), \otimes(\beta, \overline{w}))$, $\underline{u_1} = \max(\otimes(\alpha, \underline{u}), \otimes(\beta, \underline{w}))$,

$$\overline{u_2} = \max\big(\otimes(\alpha, \bar{v}), \otimes(\beta, \overline{w})\big) \quad \text{and} \quad \underline{u_2} = \max\big(\otimes(\alpha, \underline{v}), \otimes(\beta, \underline{w})\big).$$

Suppose that $L \succcurlyeq_{PU} L'$. We have to prove that $L_1 \succcurlyeq_{PU} L_2$. Recall that $\max(\alpha, \beta) = 1$ and that $L \succcurlyeq_{PU} L'$ arises in 3 cases (i.e. (i) $\bar{u} = \bar{v} = 1$ and $\underline{u} \leqslant \underline{v}$, (ii) $\bar{u} \geqslant \bar{v}$ and $\underline{u} = \underline{v} = 1$, (iii) $\bar{u} = 1$, $\bar{v} < 1$ and $\underline{v} = 1$) which corresponds to 6 different cases. We show that $L_1 \succcurlyeq_{PU} L_2$ can be deduced in each of these cases:

- Case 1: $\bar{u} = \bar{v} = 1$ and $\underline{u} \leqslant \underline{v}$, $\alpha = 1$. Then:

  $$\overline{u_1} = \max\big(\otimes(1,1), \otimes(\beta, \overline{w})\big) = 1 \quad \text{and} \quad \overline{u_2} = \max(\otimes(1,1), \otimes(\beta, \overline{w})) = 1.$$

  Thus $L_1 \succcurlyeq_{PU} L_2$ iff $\underline{u_1} \leqslant \underline{u_2}$.
  $\alpha = 1$ implies that $\underline{u_1} = \max(\underline{u}, \otimes(\beta, \underline{w}))$ and $\underline{u_2} = \max(\underline{v}, \otimes(\beta, \underline{w}))$.

  $$\underline{u} \leqslant \underline{v} \quad \text{thus implies} \quad \underline{u_1} \leqslant \underline{v_1}.$$

  Hence $L_1 \succcurlyeq_{PU} L_2$.
- Case 2: $\bar{u} = \bar{v} = 1$ and $\underline{u} \leqslant \underline{v}$, $\beta = 1$.
  Then

  $$\overline{u_1} = \max\big(\otimes(\alpha, 1), \otimes(1, \overline{w})\big) = \max(\alpha, \overline{w}) \quad \text{and}$$
  $$\overline{u_2} = \max\big(\otimes(\alpha, 1), \otimes(1, \overline{w})\big) = \max(\alpha, \overline{w}) = \overline{u_1}.$$

  Thus $L_1 \succcurlyeq_{PU} L_2$ iff $\underline{u_1} \leqslant \underline{u_2}$.

  $$\beta = 1 \quad \text{implies that} \quad \underline{u_1} = \max\big(\otimes(\alpha, \underline{u}), \underline{w}\big) \quad \text{and} \quad \underline{u_2} = \max\big(\otimes(\alpha, \underline{v}), \underline{w}\big);$$

  since $\underline{u} \leqslant \underline{v}$, we get $\underline{u_1} \leqslant \underline{u_2}$. Hence $L_1 \succcurlyeq_{PU} L_2$.
- Case 3: $\underline{u} = \underline{v} = 1$ and $\bar{u} \geqslant \bar{v}$, $\alpha = 1$. This case is similar to case 1 (exchanging the roles of positive and negative utilities).
- Case 4: $\underline{u} = \underline{v} = 1$ and $\bar{u} \geqslant \bar{v}$, $\beta = 1$. This case is similar to case 2 (exchanging the roles of positive and negative utilities).

- Case 5: $\bar{u} = 1$, $\bar{v} < 1$, $\underline{v} = 1$, $\alpha = 1$. Then:

  $$\begin{aligned}&\overline{u_1} = \max\bigl(\otimes(1,1), \otimes(\beta, \overline{w})\bigr) = 1 \quad \text{and}\\ &\underline{u_2} = \max\bigl(\otimes(1,1), \otimes(\beta, \underline{w})\bigr) = 1.\end{aligned}$$

  That is to say $\overline{u_1} = 1 \geqslant \overline{u_2}$ and $\underline{u_1} \leqslant \underline{u_2} = 1$, i.e. $L_1 = \langle 1/\top, \underline{u_1}/\bot\rangle$ and $L_2 = \langle \overline{u_2}/\top, 1/\bot\rangle$. Thus using Eq. (9), $L_1 \succcurlyeq_{PU} L_2$.
- Case 6: $\bar{u} = 1$, $\bar{v} < 1$, $\underline{v} = 1$, $\beta = 1$. Then:

  $$\begin{aligned}&\overline{u_1} = \max\bigl(\otimes(\alpha,1), \otimes(1, \overline{w})\bigr) = \max(\alpha, \overline{w}),\\ &\underline{u_1} = \max\bigl(\otimes(\alpha,\underline{u}), \otimes(1, \underline{w})\bigr) = \max\bigl(\otimes(\alpha,\underline{u}), \underline{w}\bigr),\\ &\overline{u_2} = \max\bigl(\otimes(\alpha,\bar{v}), \otimes(1, \overline{w})\bigr) = \max\bigl(\otimes(\alpha,\bar{v}), \overline{w}\bigr),\\ &\underline{u_2} = \max\bigl(\otimes(\alpha,1), \otimes(1, \underline{w})\bigr) = \max(\alpha, \underline{w}).\end{aligned}$$

  Since $\otimes$ is decreasing, $\max(\otimes(\alpha,\underline{u}), \underline{w}) \leqslant \max(\alpha, \underline{w})$ (i.e. $\underline{u_1} \leqslant \underline{u_2}$) and $\max(\alpha, \overline{w}) \geqslant \max(\otimes(\alpha, \bar{v}), \overline{w}))$ (i.e. $\overline{u_1} \geqslant \overline{u_2}$). Thus using Eq. (9), $L_1 \succcurlyeq_{PU} L_2$.

So, in any case, $L \succcurlyeq_{PU} L'$ implies that $L_1 \succcurlyeq_{PU} L_2$.

– *Weak monotonicity of U_{pes} and U_{opt}*

Consider any three lotteries L, L', L'' and let $L_1 = \mathit{Reduction}(\langle\alpha/L, \beta/L''\rangle)$ and $L_2 = \mathit{Reduction}(\langle\alpha/L', \beta/L''\rangle)$. We can, without loss of generality, suppose that L, L' and L'' are constant lotteries (thanks to certainty equivalence axiom, see [7]), i.e. $L = \langle 1/u\rangle$, $L' = \langle 1/u'\rangle$ and $L'' = \langle 1/u''\rangle$: in L (resp. L', resp. L''), any utility degree different fro u (resp. u', resp. u'') receives a possibility degree equal to 0. Then, for both interpretations of $\otimes$:

$$L_1 = \langle\alpha/u, \beta/u''\rangle \quad \text{and} \quad L_2 = \langle\alpha/u', \beta/u''\rangle.$$

Let us first show that $L \succcurlyeq_{U_{pes}} L'$ implies $L_1 \succcurlyeq_{U_{pes}} L_2$.

- If $L \sim_{U_{pes}} L'$, i.e. $u = u'$, L_1 and L_2 are identical, thus $L_1 \sim_{U_{pes}} L_2$.
- If $L \succ_{U_{pes}} L'$, we get $u > u'$ since $U_{pes}(L) = u$ and $U_{pes}(L') = u'$.

Let $a = \max(u'', 1-\beta)$, then: $U_{pes}(L_1) = \min(\max(u, 1-\alpha), a)$ and $U_{pes}(L_2) = \min(\max(u', 1-\alpha), a)$. $u > u'$ implies $\max(u, 1-\alpha) > \max(u', 1-\alpha)$. Thus $U_{pes}(L_1) \geqslant U_{pes}(L_2)$. We show in the same manner that $L \succcurlyeq_{U_{opt}} L'$ implies $L_1 \succcurlyeq_{U_{opt}} L_2$.

- If $L \sim_{U_{opt}} L'$, i.e. $u = u'$, L_1 and L_2 are identical, thus $L_1 \sim_{U_{opt}} L_2$.
- If $L \succ_{U_{opt}} L'$, we get $u > u'$ since $U_{opt}(L) = u$ and $U_{opt}(L') = u'$.

Let $a = \max(u'', 1-\beta)$, then: $U_{opt}(L_1) = \max(\min(u, 1-\alpha), a)$ and $U_{opt}(L_2) = \max(\min(u', 1-\alpha), a)$. $u > u'$ implies $\min(u, 1-\alpha) > \min(u', 1-\alpha)$. Thus $U_{opt}(L_1) \geqslant U_{opt}(L_2)$. □

Proof of Proposition 6. In what follows we develop a formal proof for Proposition 6 which is a direct consequence of the basic axiom defined by Dubois et al. [3]. Consider any three lotteries L, L' and L''. We can suppose without loss of generality that they are in a reduced form, i.e.:

$$L = \langle\lambda_1/u_1, \ldots, \lambda_n/u_n\rangle, \qquad L' = \bigl\langle\lambda'_1/u_1, \ldots, \lambda'_n/u_n\bigr\rangle \quad \text{and} \quad L'' = \bigl\langle\lambda''_1/u_1, \ldots, \lambda''_n/u_n\bigr\rangle.$$

Let $L_1 = \mathit{Reduction}(\langle\alpha/L, \beta/L''\rangle)$ and $L_2 = \mathit{Reduction}(\langle\alpha/L', \beta/L''\rangle)$ with $\max(\alpha, \beta) = 1$. The possibility of getting a utility degree u_i from L_1 (resp. L_2) is equal to $\lambda^1_i = \max(\otimes(\alpha, \lambda_i), \otimes(\beta, \lambda''_i))$ (resp. $\lambda^2_i = \max(\otimes(\alpha, \lambda'_i), \otimes(\beta, \lambda''_i)))$.

When $\alpha = 1$, we have

$$\lambda^1_k = \max\bigl(\lambda_i, \otimes(\beta, \lambda''_i)\bigr) \quad \text{and} \quad \lambda^2_i = \max\bigl(\lambda'_i, \otimes(\beta, \lambda''_i)\bigr).$$

When $\beta = 1$, we have

$$\lambda^1_i = \max\bigl(\otimes(\alpha, \lambda_i), \lambda''_i\bigr) \quad \text{and} \quad \lambda^2_i = \max\bigl(\otimes\bigl(\alpha, \lambda'_i\bigr), \lambda''_i\bigr).$$

Let $U_\top = \{u_i \colon \lambda_i = 1\}$ (resp. $U'_\top = \{u_i \colon \lambda'_i = 1\}$, $U^1_\top = \{u_i \colon \lambda^1_i = 1\}$, $U^2_\top = \{u_i \colon \lambda^2_i = 1\}$) the set of utility degrees receiving a possibility equal to 1 in L (resp. L', L_1, L_2). These sets are not empty since the distributions are normalized. It holds that $L \succcurlyeq_{L\Pi} L'$, i.e. $\Pi(L \succcurlyeq L') \geqslant \Pi(L' \succcurlyeq L)$ if and only if $\max_{u_i \in U_\top} \geqslant \min_{u_i \in U'_\top}$ [3]. Let us show that $L \succcurlyeq_{L\Pi} L'$ implies $L_1 \succcurlyeq_{L\Pi} L_2$.

- If $\alpha = 1$: we have $U_\top \subseteq U^1_\top$ and $U'_\top \subseteq U^2_\top$. Hence $\max_{u_i \in U_\top}$ belongs to $U^1_\top$ and $\min_{u_i \in U'_\top}$ belongs to $U^2_\top$, $\max_{u_i \in U^1_\top} \geqslant \max_{u_i \in U_\top}$ and $\min_{u_i \in U^2_\top} \leqslant \min_{u_i \in U'_\top}$.

  Thus $\max_{u_i \in U^1_\top} \geqslant \min_{u_i \in U^2_\top}$.

- If $\alpha < 1$ and $\beta = 1$: let u_i be any of the degrees that receive a possibility degree equal to 1 in L''. Since $\beta = 1$, u_i belongs to both $U^1_\top$ and $U^2_\top$. Thus $\Pi(L_1 \succcurlyeq L_2) = \Pi(L_2 \succcurlyeq L_1) = 1$.

Thus $L \succcurlyeq_{L\Pi} L'$ implies that $L_1 \succcurlyeq_{L\Pi} L_2$ in both cases. □

Proof of Proposition 7. We give the proof for $\forall L, L' \in \mathcal{L}, L \succcurlyeq_{L\Pi} L' \iff L^{core} \succcurlyeq_{L\Pi} L'$. (The proof of $\forall L, L' \in \mathcal{L}, L' \succcurlyeq_{L\Pi} L \iff L' \succcurlyeq_{L\Pi} L^{core}$ is similar.)

$L \succcurlyeq_{L\Pi} L' \iff \Pi(L \succcurlyeq L') \geqslant \Pi(L' \succcurlyeq L)$. Because the possibility distributions are normalized, $max(\Pi(L \succcurlyeq L') \geqslant \Pi(L' \succcurlyeq L) = 1)$:

$$L \succcurlyeq_{L\Pi} L' \quad \text{iff} \quad \Pi\big(L \succcurlyeq L'\big) = 1.$$

$\Pi(L \succcurlyeq L') = 1$ if there exists $u \in L_{=1}$ and $u' \in L'_{=1}$ such that $u \geqslant u'$. This condition is equivalent to $\max(L_{=1}) \geqslant u'$.

Then $L \succcurlyeq_{L\Pi} L'$ implies $L^{core} \succcurlyeq_{L\Pi} L'$ (the core of L^{core} precisely contains $\max(L_{=1})$). Reciprocally, if $L^{core} \succcurlyeq_{L\Pi} L'$, then $\exists u'$ s.t. $\max(L_{=1}) \geqslant u'$, and thus $L \succcurlyeq_{L\Pi} L'$. □

Proof of Proposition 8. Consider any three lotteries L, L' and L''. Let $L_1 = Reduction(\langle \alpha/L, \beta/L''\rangle)$ and $L_2 = Reduction(\langle \alpha/L', \beta/L''\rangle)$ with $\max(\alpha, \beta) = 1$.

Recall that $L \succcurlyeq_{LN} L'$ means that $\Pi(L \succ L') \geqslant \Pi(L' \succ L)$. When L and L' are unimodal and share the same mode, L^{2nd} is of the form $\langle \gamma/m, 1/u, \gamma/M\rangle$ and L'^{2nd} is of the form $\langle \gamma'/m', 1/u, \gamma'/M'\rangle$; then $\Pi(L \succ L') \leqslant \max(\gamma, \gamma')$, $\Pi(L \prec L') \leqslant \max(\gamma, \gamma')$ and $L \succcurlyeq_{LN} L'$ iff either ($\gamma > \gamma'$ and $M > u$) or ($\gamma' > \gamma$ and $u > m'$) or ($\gamma = \gamma'$ and $\max(u, M) > \min(u, m')$).

Let us first assume that $\alpha = 1$, then two cases can be distinguished:

- Case 1: $\max(L_{=1}) > \min(L'_{=1})$ (and then $\Pi(L \succ L') = 1$, which implies $L \succcurlyeq_{LN} L'$); since $\alpha = 1$, $\max(L_{=1})$ belongs to the core of L_1 and $\min(L'_{=1})$ belongs to the core of L_2; thus $\Pi(L_1 \succ L_2) = 1$ which implies $L_1 \succcurlyeq_{LN} L_2$.
- Case 2: $\min(L'_{=1}) \geqslant \max(L_{=1})$. Then the two lotteries are unimodal and share the same mode, say u (if this was not the case, we had $\Pi(L' \succ L) = 1$ and $\Pi(L \succ L') < 1$, which contradicts $L \succcurlyeq_{LN} L'$). L^{2nd} is of the form $\langle \gamma/m, 1/u, \gamma/M\rangle$ and L'^{2nd} is of the form $\langle \gamma'/m', 1/u, \gamma'/M'\rangle$. Since $\alpha = 1$, the core of L_1^{2nd} and L_2^{2nd} contains u (and only u).

- If $\beta > \max(\gamma, \gamma')$, $L_1^{2nd} = L_2^{2nd}$ (value u at level $\pi = 1$ and bounds of the core of L'' at level $\pi = \beta$), and thus $L_1 \sim_{LN} L_2$.
- If $\beta < \min(\gamma, \gamma')$, $L_1^{2nd} = L^{2nd}$ and $L_2^{2nd} = L'^{2nd}$ and thus $L \succcurlyeq_{LN} L'$ implies $L_1 \succcurlyeq_{LN} L_2$.
- If $\gamma \geqslant \beta \geqslant \gamma'$, L_1^{2nd} contains u at level $\pi = 1$, a value equal or greater than M at level γ and the core of L_2^{2nd} contains u. Moreover $L \succcurlyeq_{LN} L'$ implies that $M > u$ (otherwise we had $\Pi(L \succ L') < \gamma$ and $\Pi(L' \succ L) = \gamma$). Thus $L_1 \succcurlyeq_{LN} L_2$.
- If $\gamma' \geqslant \beta \geqslant \gamma$, L_2^{2nd} contains u at level $\pi = 1$, a value equal or lower than m' at level γ' and the core of L_1^{2nd} contains u. Moreover $L \succcurlyeq_{LN} L'$ implies that $u > m'$ (otherwise we had $\Pi(L \succ L') < \gamma$ and $\Pi(L' \succ L) = \gamma$). Thus $L_1 \succcurlyeq_{LN} L_2$.

When $\alpha < 1$ (and thus $\beta = 1$) the core of L_1 and L_2 is the one of L'' and two cases can be distinguished:

- Case 1: L'' is not unimodal, the comparison is made at level $\pi = 1$ and thus $L_1 \sim_{LN} L_2$;
- Case 2: L'' is unimodal, L''^{2nd} is of the of the form $\langle \gamma''/m'', 1/u'', \gamma''/M''\rangle$; because L and L' are also unimodal, so are L_1 and L_2 – because $\alpha < 1$, their mode is u''. Thus:
  - If $\alpha < \gamma''$, then $L_1^{2nd} = L_2^{2nd} = L''^{2nd}$: $L_1 \sim_{LN} L_2$;
  - If $\alpha \geqslant \gamma''$ and $u \neq u''$, then the second level on L_1 (resp. L_2) contains the core of L_1 (resp. L_2), i.e. contains u. Thus L_1^{2nd} is of the form $\langle \alpha/v, 1/u'', \alpha/V\rangle$ with $v \leqslant u$ and $V \geqslant u$ and L_2^{2nd} is of the form $\langle \alpha/v', 1/u'', \alpha/V'\rangle$ with $v' \leqslant u$ and $V' \geqslant u$;
    If $u > u''$, then $\Pi(L_1 \succ L_2) = \alpha$ and $\Pi(L_2 \succ L_1) = \alpha$;
    If $u < u''$, then $\Pi(L_1 \succ L_2) = \alpha$ and $\Pi(L_2 \succ L_1) = \alpha$. In both cases $L_1 \sim_{LN} L_2$;
  - If $u = u''$ and $\gamma'' \geqslant \max(\min(\alpha, \gamma'), \min(\alpha, \gamma')$ then the second level on L_1 (resp. L_2) is the one of possibility degree $\pi = \gamma''$ and contains the second level of L'':
    L_1^{2nd} is of the form $\langle \gamma''/v, 1/u'', \gamma''/V\rangle$ with $v \leqslant m''$ and $V \geqslant M''$ and L_2^{2nd} is of the form $\langle \gamma''/v', 1/u'', \gamma''/V'\rangle$ with $v' \leqslant m''$ and $V' \geqslant M''$. Since $M'' \neq u''$ and $m'' \neq u''$, $\Pi(L_2 \succ L_1) = \Pi(L_1 \succ L_2) = \gamma''$; $L_1 \sim_{LN} L_2$;
  - If $u = u''$ and $\max(\min(\alpha, \gamma), \min(\alpha, \gamma')) \geqslant \gamma''$ then the second level on L_1 (resp. L_2) is the one of possibility degree $\pi = \min(\alpha, \gamma)$ (resp. $\pi = \min(\alpha, \gamma')$) and contains the second level of L (resp. L'): L_1^{2nd} is of the form $\langle \min(\alpha, \gamma)/v, 1/u, \min(\alpha, \gamma)/V\rangle$ with $v \leqslant m$ and $V \geqslant M$, and L_2^{2nd} is of the form $\langle \min(\alpha, \gamma')/v', 1/u, \min(\alpha, \gamma')/V'\rangle$ with $v' \leqslant m'$ and $V' \geqslant M'$.
    If $L \succcurlyeq_{LN} L'$ because $M > u$ and $\gamma \geqslant \gamma'$, then $V > u$ and thus $\Pi(L_1 \succ L_2) = \min(\alpha, \gamma)$ and $\Pi(L_1 \prec L_2) \leqslant \min(\alpha, \gamma)$: $L_1 \succcurlyeq_{LN} L_2$; If $L \succcurlyeq_{LN} L'$ because $u > m'$ and $\gamma' \geqslant \gamma$, then $u > v'$ and thus $\Pi(L_1 \succ L_2) = \min(\alpha, \gamma')$ and $\Pi(L_1 \prec L_2) \leqslant \min(\alpha, \gamma')$: $L_1 \succcurlyeq_{LN} L_2$;
  - If $u = u''$ and $\min(\alpha, \gamma) > \gamma'' > \min(\alpha, \gamma')$ then (i) $\gamma > \gamma'$ and then $L \succcurlyeq_{LN} L'$ implies $M > u$ and (ii) the second level on L_1 is the one of possibility degree $\pi = \min(\alpha, \gamma)$ contains the second level of L: L_1^{2nd} is of the form

$\langle \min(\alpha, \gamma)/v, 1/u, \min(\alpha, \gamma)/V \rangle$ with $V \geqslant M$: from $M > u$, we get $\Pi(L_1 \succ L_2) = \min(\alpha, \gamma)$. Since $\Pi(L_2 \succ L_1) \leqslant \min(\alpha, \gamma)$: $L_1 \succcurlyeq_{LN} L_2$;

- If $u = u''$ and $\min(\alpha, \gamma') > \gamma'' > \min(\alpha, \gamma)$ then (i) $\gamma' > \gamma$ and then $L \succcurlyeq_{LN} L'$ implies $u > m'$ and (ii) the second level on L_2 is the one of possibility degree $\pi = \min(\alpha, \gamma')$ and contains the second level of L': L_2^{2nd} is of the form $\langle \min(\alpha, \gamma')/v, 1/u, \min(\alpha, \gamma')/V \rangle$ with $v \leqslant m'$; from $u > m'$ we get $\Pi(L_1 \succ L_2) = \min(\alpha, \gamma')$. Since $\Pi(L_2 \succ L_1) \leqslant \min(\alpha, \gamma')$: $L_1 \succcurlyeq_{LN} L_2$. □

Proof of Proposition 9. We give the proof for $\forall L, L' \in \mathcal{L}, L \succcurlyeq_{LN} L' \iff L^{2nd} \succcurlyeq_{LN} L'$. (The proof of $L' \succcurlyeq_{LN} L \iff L' \succcurlyeq_{LN} L^{2nd}$ is similar.)

$L \succcurlyeq_{LN} L'$ iff $1 - \Pi(L' \succ L) \geqslant 1 - \Pi(L \succ L')$, which is equivalent to $\Pi(L \succ L') \geqslant \Pi(L' \succ L)$. This inequality is satisfied in three cases only:

- $\max(L_{=1}) > \min(L'_{=1})$ (then $\Pi(L \succ L') = 1$).
- $\max(L_{=1}) = \min(L'_{=1}) = \max(L'_{=1}) = \min(L_{=1})$ (let u denote this value), $\alpha^{2nd} \geqslant \alpha'^{2nd}$ and $\max(L_{=\alpha^{2nd}}) > u$ (then $\Pi(L \succ L') = \alpha^{2nd}$ and $\Pi(L' \succ L) \leqslant \alpha^{2nd}$).
- $\max(L_{=1}) = \min(L'_{=1}) = \max(L'_{=1}) = \min(L_{=1})$ (let u denote this value), $\alpha'^{2nd} \geqslant \alpha^{2nd}$ and $u > \min(L'_{=\alpha'^{2nd}})$.

Since L^{2nd} precisely contains $\max(L_{=1})$ and $\min(L_{=1})$ (at level 1) and $\max(L_{=\alpha^{2nd}})$ (at level α^{2nd}), $L \succcurlyeq_{LN} L' \Rightarrow L^{2nd} \succcurlyeq L'$.

Reciprocally, if $L^{2nd} \succcurlyeq L'$, then it is easy to check that one of the three previous cases is fulfilled and thus $L \succcurlyeq_{LN} L'$. □

Proof of Proposition 10. Let $L = \langle 1/u, \alpha/m, \alpha/M \rangle$ and $L' = \langle 1/u', \alpha'/m', \alpha'/M' \rangle$, with $\alpha \neq \alpha'$. (Similar results hold for $\alpha' > \alpha$.)

- If $u \neq u'$, then only the modes are used to make the decision: either $u > u'$, and then $L \succ_{LN} L'$, i.e. $\Pi(L > L') = 1$ and $\Pi(L' < L) < 1$; or $u' > u$ and then $L' \succ_{LN} L$, i.e. $\Pi(L > L') < 1$ and $\Pi(L' < L) = 1$.
- If $u = u'$, decision is made at level $\max(\alpha, \alpha')$; when $\alpha > \alpha'$, then either $u > M$ (and then $L' \succ_{LN} L$); or $u < m$ (and then $L \succ_{LN} L'$); or $M > u > m$ (and then $L \sim_{LN} L'$).

So, $u = u'$ is a necessary condition for the indifference of the two lotteries.

Moreover, $M = u$ (resp. $m = u$) is not possible since the possibility of u is equal to 1 for both L and L' and the possibility of M (resp. m) is strictly lower than 1. Thus two cases are possible:

- Case 1: $M = m$, then the two possible cases are:
  - If $u > M = m$, then $\Pi(L' > L) = \alpha$ (at level α, L' can take value u and L value m) and $\Pi(L > L') < \alpha$ (one have go below α to find a value for L strictly greater than u); $\Pi(L > L')$ cannot be equal to $\Pi(L' > L')$, i.e. L and L' cannot be indifferent.
  - If $u < M = m$, then $\Pi(L > L') = \alpha$ (at level α, L' can take value u and L value m) and $\Pi(L' > L) < \alpha$ (one have go below α to find a value for L strictly lower than u); i.e. L and L' cannot be indifferent.
- Case 2: $M \neq m$, then by construction $M > m$, i.e. three cases are possible:
  - If $u > M$, then $\Pi(L' > L) = \alpha$ (at level α, L' can take value u and L value M) and $\Pi(L > L') < \alpha$ (one have go below α to find a value for L strictly greater than u). So $L \sim_{LN} L'$ is not satisfied.
  - If $u < m$, then $\Pi(L > L') = \alpha$ (at level α, L' can take value u and L value m) and $\Pi(L' > L) < \alpha$ (one have go below α to find a value strictly lower than u). So $L \sim_{LN} L'$ implies $m \leqslant u$. $L \sim_{LN} L'$ is not satisfied.
  - If $M > u > m$, then $\Pi(L > L') = \alpha = \Pi(L' > L)$.

  Hence, when $M > m$, $L \sim_{LN} L'$ implies $M > u > m$.

  Reciprocally, it is easy to check that $M > u > m$ and $\alpha > \alpha'$ imply $\Pi(L > L') = \alpha$ and $\Pi(L' > L) = \alpha$. □

Proof of Proposition 11(i). We have $L_1 = \langle \gamma/L, \beta/L'' \rangle$ and $L_2 = \langle \gamma/L', \beta/L'' \rangle$ where $L = \langle 1/u, \alpha/m, \alpha/M \rangle$ and $L' = \langle 1/u, \alpha'/m', \alpha'/M' \rangle$ are two indifferent 2nd core lotteries (i.e. $L \sim_{LN} L'$), $\alpha < \alpha'$ and $\max(\gamma, \beta) = 1$.

Thanks to Proposition 9 we can assume without loss of generality that L_1, L_2 and L_b are 2nd core lotteries (if it is not the case, simply replace them by their 2nd core lotteries). Since the cores of L and L' are equal, the cores of L_1 and L_2 are also equal. So L_1 is unimodal iff L_2 is unimodal (and both share the same mode).

- If L_b is not unimodal, then the decision is made at possibility level 1, where L_1 and L_2, are identical. Hence the property.
- If L_1 and L_2 are not unimodal, then the decision is made at possibility level 1, where L_1 and L_2, are identical. Hence the property.

- If the three lotteries are unimodal but the mode of L_1 and L_2 is not equal to the one of L_b, then the decision is made at possibility level 1, where L_1 and L_2 are identical. Hence the property.
- If L_b, L_1 and L_2 are unimodal and share the same mode, u.
  L_1 can be written as $\langle 1/u, a_1/m_1, a_1/M_1\rangle$, with $m_1 < M_1$ and $a_1 < 1$,
  L_2 can be written as $\langle 1/u, a_2/m_2, a_2/M_2\rangle$, with $m_2 < M_2$ and $a_2 < 1$,
  L_b can be written as $\langle 1/u, b/m_b, b/M_b\rangle$, with $m_b < M_b$ and $b < 1$.
  Notice that $u \notin \{M_1, m_1, M_2, m_2, M_b, m_b\}$ (the possibility degree of u is equal to 1 while the possibility degrees of M_1, m_1, M_2, m_2, M_b and m_b are equal to a_1, a_2 or b which are all strictly less than 1).
  Since $\alpha < \alpha'$, $a_1 < a_2$. Moreover, LN satisfies the principle of weak monotonicity, thus $L \sim_{LN} L'$ implies $L_1 \sim_{LN} L_2$. Thanks to Proposition 10, $L_1 \sim_{LN} L_2$ and $a_1 < a_2$ imply $m_2 < u < M_2$.
  By hypothesis, $L_2 \succ_{LN} L_b$. We have $m_2 < u < M_2$, suppose that $a_2 \geqslant b$, this implies $\Pi(L_2 > L_b) = a_2$ and $\Pi(L_b > L_2) = a_2$. Thus $a_2 \geqslant b$ contradicts the hypothesis. Hence $b > a_2$.
  Since $b > a_2$, $\Pi(L_2 > L_b) > \Pi(L_b > L_2)$ iff $\Pi(L_2 > L_b) = b$ and $\Pi(L_b > L_2) < b$. Which means that at level b, $u > M_b$ (if $u < M_b$, $\Pi(L_b > L_2) = b$ and $u = M_b$ is not possible).
  $a_1 < b$ (Since $a_1 < a_2$ and $a_2 < b$): the comparison of L_1 and L_b is made at level b. From $u > M_b$ we conclude $\Pi(L_1 > L_b) = b$ and $\Pi(L_b > L_1) < b$, i.e. $L_1 \succ_{LN} L_b$. Hence when L_b, L_1 and L_2 are unimodal and share the same mode, $L_2 \succ_{LN} L_b$ implies $L_1 \succ_{LN} L_b$. □

Proof of Proposition 11(ii). The proof is similar to the previous one. Thanks to Proposition 9 we can assume without loss of generality that L_1, L_2 and L_b are 2nd core lotteries. Consider the case where they are unimodal and share the same mode, u (if not, refer to the previous proof). Then:

L_1 can be written as $\langle 1/u, a_1/m_1, a_1/M_1\rangle$, with $m_1 < M_1$ and $a_1 < 1$,
L_2 can be written as $\langle 1/u, a_2/m_2, a_2/M_2\rangle$, with $m_2 < M_2$ and $a_2 < 1$,
L_b can be written as $\langle 1/u, b/m_b, b/M_b\rangle$, with $m_b < M_b$ and $b < 1$.

Following the previous proof, we have $u \notin \{M_1, m_1, M_2, m_2, M_b, m_b\}$ and also $L_1 \sim_{LN} L_2$ and $a_1 < a_2$ imply $m_2 < u < M_2$.

By hypothesis, $L_b \succ_{LN} L_2$. But $a_2 \geqslant b$ and $m_2 < u < M_2$ imply that $\Pi(L_2 > L_b) = a_2$ and $\Pi(L_b > L_2) = a_2$. Thus $a_2 \geqslant b$ contradicts the hypothesis. Hence $b > a_2$.

Since $b > a_2$, $\Pi(L_2 < L_b) > \Pi(L_b < L_2)$ iff $\Pi(L_2 < L_b) = b$ and $\Pi(L_b < L_2) < b$. Which means that at level b, $u < m_b$ (if $u > m_b$, $\Pi(L_b < L_2) = b$ and $u = m_b$ is impossible).

$a_1 < b$ (Since $a_1 < a_2$ and $a_2 < b$): the comparison of L_1 and L_b is made at level b. From $u < m_b$ we conclude that $\Pi(L_1 < L_b) = b$ and $\Pi(L_b < L_1) < b$, i.e. $L_1 \succ_{LN} L_1$. Hence, when L_1, L_2 and L_b are unimodal and share the same mode, $L_b \succ_{LN} L_2$ implies $L_b \succ_{LN} L_1$. □

Proof of Proposition 13. Let L, L' and L'' be 3 kappa lotteries. We can suppose without loss of generality that they are in reduced form, i.e. that:

$$L = \langle \kappa_1/\mu_1, \ldots, \kappa_n/\mu_m\rangle, \qquad L' = \langle \kappa'_1/\mu_1, \ldots, \kappa'_n/\mu_m\rangle \quad \text{and} \quad L'' = \langle \kappa''_1/\mu_1, \ldots, \kappa''_n/\mu_m\rangle.$$

Let $L_1 = \mathit{Reduction}(\langle \alpha/L, \beta/L''\rangle)$ and $L_2 = \mathit{Reduction}(\langle \alpha/L', \beta/L''\rangle)$.

According to Eq. (22) the kappa-ranking of utility degree u^1_k from L_1 is

$$\kappa^1_k = \min\bigl((\alpha + \kappa_k), \bigl(\beta + \kappa''_k\bigr)\bigr).$$

Thus:

$$\mathit{OMEU}(L_1) = \min_{i=1..m} \bigl(\min\bigl[(\kappa_i + \alpha), \bigl(\kappa''_i + \beta\bigr)\bigr] + u_i\bigr).$$

Similarly:

$$\mathit{OMEU}(L_2) = \min_{i=1..m} \bigl(\min\bigl[\bigl(\kappa'_i + \alpha\bigr), \bigl(\kappa''_i + \beta\bigr)\bigr] + u_i\bigr).$$

Suppose that $\mathit{OMEU}(L) \leqslant \mathit{OMEU}(L')$, i.e.

$$\begin{aligned} &\min_{i=1..m} \{\kappa_i + u_i\} \leqslant \min_{i=1..m} \bigl\{\kappa'_i + u_i\bigr\} \\ &\quad\Rightarrow \quad \min_{i=1..m} \{\kappa_i + u_i\} + \alpha \leqslant \min_{i=1..m} \bigl\{\kappa'_i + u_i\bigr\} + \alpha \\ &\quad\Rightarrow \quad \min_{i=1..m} \{\kappa_i + u_i + \alpha\} \leqslant \min_{i=1..m} \bigl\{\kappa'_i + u_i + \alpha\bigr\}. \end{aligned}$$

As a consequence, we get:

$$\min\Big(\min_{i=1..m}\{\kappa_i''+u_i+\beta\}, \min_{i=1..m}\{\kappa_i+u_i+\alpha\}\Big)$$
$$\leqslant \min\Big(\min_{i=1..m}\{\kappa_i''+u_i+\beta\}, \min_{i=1..m}\{\kappa_i'+u_i+\alpha\}\Big).$$

By associativity of the min operator, we get:

$$\min_{i=1..m}\min\big(\{\kappa_i''+u_i+\beta\},\{\kappa_i+u_i+\alpha\}\big)\leqslant \min_{i=1..m}\min\big(\{\kappa_i''+u_i+\beta\},\{\kappa_i'+u_i+\alpha\}\big).$$

Hence: $\min_{i=1..m}\min[(\kappa_i+\alpha),(\kappa_i''+\beta)]+u_i \leqslant \min_{i=1..m}\min[(\kappa_i'+\alpha),(\kappa_i''+\beta)]+u_i$.

That is to say that $OMEU(L_1)\leqslant OMEU(L_2)$. □

Proof of Proposition 14. We first show that DT-OPT-Ch_N and DT-OPT-Ch_Π belong to the NP class. Next, we prove that 3CNF can be polynomially reduced to our problem, following the principle used by [16] for proving the difficulty of the problem in the RDU case. For DT-OPT-Ch_Π, the transformation we used is a simplification of the one proposed in [16]. For DT-OPT-Ch_N, the underlying idea is different and is mainly based on the fact that compounding a lottery with another lottery that is not strictly better than the original one cannot increase its Choquet value (see Section 2.6).

Membership to NP

The membership of DT-OPT-Ch_N (resp DT-OPT-Ch_Π) to NP is straightforward, using the following procedure:

```
1. Call to the NP oracle
   • Guess a strategy $\delta$ from T
2. Polytime checking
   • Compute $L = Reduce(\delta)$, using Eq. (5)
3. Compute $Ch_\mu$ ($\mu$ is $N$ or $\Pi$)
   • $\max \leftarrow 0$
   • $Ch_\mu \leftarrow u_1$
   • If $\mu = N$, then $\forall i = 1, m-1$
     – if $\max < \lambda_j^L$, then $\max \leftarrow \lambda_j^L$
     – $Ch_\mu \leftarrow Ch_\mu + (u_{i+1} - u_i) * (1 - \max)$
   • If $\mu = \Pi$, then $\forall i = m, 2$
     – If $\max < \lambda_j^L$, then $\max \leftarrow \lambda_j^L$
     – $Ch_\mu \leftarrow Ch_\mu + (u_i - u_{i-1}) * \max$
   • Check that $Ch_\mu \geqslant \theta$
```

Since the reduction operation is linear in the size of the compound lottery and the computation of the necessity-based Choquet value (resp. the possibility-based Choquet value) is linear in the number of utility levels in the utility scale, the full procedure is polynomial. Hence DT-OPT-Ch_N (resp. DT-OPT-Ch_Π) belongs to NP.

NP-hardness of DT-OPT-Ch_N

The hardness of DT-OPT-Ch_N is obtained by a reduction from the 3-SAT problem. Each instance of the 3-SAT problem is a CNF-formula consisting of m clauses, denoted by $Cl = \{Cl_1, \ldots, Cl_m\}$, on a set of n propositional variables $X = \{X_1, \ldots, X_n\}$ such that $\forall X_i \in X$, x_i and $\neg x_i$ are called literals. The set of all literals is denoted by $\mathcal{L}$ and l_i denotes any literal in $\mathcal{L}$. Each clause Cl_i is composed of 3 literals. Given a 3CNF, the question is whether there is a value-assignment of its literals which satisfies its clauses or not. The principle of the transformation from any 3CNF into a possibilistic decision tree can be summarized as follows:

- For each literal $l_i \in \mathcal{L}$ we define a utility u_{l_i} and a possibility degree λ_{l_i}. We also define a utility degree $u_\top$ such that $\forall l_i \in \mathcal{L}$, $u_\top > u_{l_i}$.
- For each $X_i \in X$ we associate a decision node D_{X_i} with two children relative to the two choices x_i and $\neg x_i$, i.e. $C_{x_i} = \langle 1/u_\top, \lambda_{x_i}/u_{x_i}\rangle$ and $C_{\neg x_i} = \langle 1/u_\top, \lambda_{\neg x_i}/u_{\neg x_i}\rangle$.
- For each $Cl_i \in Cl$, we associate a decision node D_{Cl_i} with three children relative to its three literals. In fact, for each literal $l_i \in Cl_i$, we associate a chance node $C_{l_i}^{Cl_i} = \langle 1/u_\top, \lambda_{l_i}/u_{l_i}\rangle$ meaning that the satisfaction of Cl_i is ensured by the choice of l_i. When selecting a chance node for D_{Cl_i}, a strategy specifies how it intends to satisfy Cl_i.

More formally, given a CNF $Cl = \{Cl_1, \ldots, Cl_m\}$ on $X = \{x_1, \ldots, x_n\}$, the necessity-based transformation into a possibilistic tree $\Pi\mathcal{T}$ can be ensured by the following procedure.

```
1. Fix $\epsilon \in [0,1]$ such that $\epsilon^n < 0.5$
2. $\forall x_i \in X$
   • $u_{x_i} \leftarrow 2(n-i)+1$
   • $\lambda_{x_i} \leftarrow \epsilon^{i+1}$
   • $u_{\neg x_i} \leftarrow 2(n-i)+2$
   • $\lambda_{\neg x_i} \leftarrow \epsilon^{i}$
3. Let $u_\top \leftarrow 2n+1$
4. Create a decision node $D_0$ as the root of $\Pi\mathcal{T}$
5. Create a chance node $H$ as the unique child of $D_0$
6. $\forall X_i \in X$ Create a decision node $D_{X_i}$ with two children $C_{x_i}$ and $C_{\neg x_i}$ s.t.:
   • $C_{x_i}$ is the simple lottery $\langle 1/u_\top, \lambda_{x_i}/u_{x_i}\rangle$
   • $C_{\neg x_i}$ is the simple lottery $\langle 1/u_\top, \lambda_{\neg x_i}/u_{\neg x_i}\rangle$
   • Add $D_{X_i}$ to the children of $H$, with a possibility degree equal to 1
7. $\forall Cl_i \in Cl$
   • Create a decision node $D_{Cl_i}$
   • $\forall l_i \in Cl_i$
     - Create a chance node $C_{l_i}^{Cl_i}$ corresponding to the simple lottery $\langle 1/u_\top, \lambda_{l_i}/u_{l_i}\rangle$
     - Add $C_{l_i}^{Cl_i}$ to the children of $D_{Cl_i}$
   • Add $D_{Cl_i}$ to the children of $H$, with a possibility degree equal to 1
```

This transformation is performed in $O(m+n)$ since the resulted decision tree contains $m+n+1$ decision nodes, $3m+2n+1$ chance nodes and $(3m+2n)\times 2$ leaves.

A strategy δ can select the literals in a consistent manner (in this case, if L is chosen for x_i, $\neg l$ is never chosen for a D_{Cl_i}) or in a contradictory manner (i.e. δ selects L in some decision and $\neg l$ for some others). By construction, there is a bijection between the non-contradictory strategies, if any, and the models of the formula.

The simple lottery equivalent to some strategy δ is the following: $\pi(\top)=1$, $\pi(u_l)=\lambda_l$ if literal L is chosen for some decision node, $\pi(u_l)=0$ otherwise. The set of simple lotteries equivalent to contradictory strategies is included in: $\mathcal{L}_{NC}=\{L\colon\ L[u_\top]=1, \forall l\in L, L[u_l]\in\{0,\lambda_l\}$ and $\min(L[u_l], L[u_{\neg l}]=0\}$. The set of simple lotteries equivalent to contradictory strategies is included in: $\mathcal{L}_C=\{L\colon\ L[u_\top]=1, \forall l\in L, L[u_l]\in\{0,\lambda_l\}, \exists l\in L$ s.t. $\min(L[u_l], L[u_{\neg l}])\neq 0\}$.

The principle of the proof is to set the values of the λ_l's and the u_l's in such a way that the Choquet value of the worst of the non-contradictory lotteries is greater than the Choquet value of the best contradictory lottery. To this extend, we choose an $\epsilon\in[0,1]$ such that $\epsilon^n<0.5$.

Then we set: $\lambda_{x_i}=\epsilon^{i+1}$, $u_{x_i}=2(n-i)+1$, $\lambda_{\neg x_i}=\epsilon^i$, $u_{\neg x_i}=2(n-i)+2$ and $u_\top=2n+1$. It holds that:

- The worst non-contradictory lottery in $\mathcal{L}_{NC}$ (i.e. the one having the lowest Ch_N value), denoted by $L_{NC}^{\downarrow}$, is such as all the positive literals are possible and the possibility of any negative literal is equal to 0, i.e.[4]

  $$L_{NC}^{\downarrow}=\langle \lambda_{x_n}/u_{x_n},\dots,\lambda_{x_1}/u_{x_1},1/u_\top\rangle.$$

  This holds since according to the proposed encoding, utilities of positive literal are always lower than their negative version ($u_{x_i}=2(n-i)+1<u_{\neg x_i}=2(n-i)+2$).
- The best contradictory lottery in $\mathcal{L}_C$ (i.e. the one having the greatest Ch_N value), denoted by $L_C^{\uparrow}$ is such as all negative literals are possible and the possibility of any positive literal is equal to 0 except for x_1 (i.e. the one having the greatest utility value within all positive literals), i.e.[5]

  $$L_C^{\uparrow}=\langle \lambda_{\neg x_n}/u_{\neg x_n},\dots,\lambda_{x_1}/u_{x_1},\lambda_{\neg x_1}/u_{\neg x_1},1/u_\top\rangle.$$

  This holds since (i) according to the proposed codification, utilities of negative literal are always greater than their positive version ($u_{x_i}=2(n-i)+1<u_{\neg x_i}=2(n-i)+2$), moreover the presence of x_1 and $\neg x_1$ causes the contradiction (we choose x_1 since u_{x_1} is the greatest utility value within all positive literals) and (ii) the less the number of utilities in the lottery receiving a non-negative possibility degree, the greater the Choquet value (Proposition 2).

Considering $L_{NC}^{\downarrow}$, the utilities that receive a positive possibility degrees are, by increasing order: $u_{x_n}<u_{x_{n-1}}<\cdots<u_{x_1}<u_\top$ ($\forall i=1..n, \neg x_i$, receive a possibility degree equal to 0). Hence:

[4] For the sake of simplicity we omitted terms where possibility degrees are equal to 0.

$$
\begin{aligned}
Ch_N(L^{\downarrow}_{NC} = {} & u_{x_n} \\
& + (u_{x_{n-1}} - u_{x_n})(1 - \lambda_{x_n}) \\
& + (u_{x_{n-2}} - u_{x_{n-1}})\big(1 - \max(\lambda_{x_n}, \lambda_{x_{n-1}})\big) \\
& + \cdots \\
& + (u_{x_1} - u_{x_2})\big(1 - \max(\lambda_{x_n}, \ldots, \lambda_{x_2})\big) \\
& + (u_{\top} - u_{x_1})\big(1 - \max(\lambda_{x_n}, \ldots, \lambda_{x_2}, \lambda_{x_1})\big) \\
= {} & 1 + 2(1 - \lambda_{x_n}) + 2(1 - \lambda_{x_{n-1}}) + \cdots + 2(1 - \lambda_{x_1}) \\
= {} & 2n + 1 - 2(\lambda_{x_n} + \cdots + \lambda_{x_1}).
\end{aligned}
$$

Considering $L^{\uparrow}_{C}$, the utilities that receive a positive degree of possibility are, by increasing order: $u_{\neg x_n} < u_{\neg x_{n-1}} < \cdots < u_{\neg x_2} < u_{x_1} < u_{\neg x_1} < u_{\top}$ (all the x_i, $i > 1$, receives a possibility degree equal to 0). Hence:

$$
\begin{aligned}
Ch_N\big(L^{\uparrow}_{C}\big) = {} & u_{\neg x_n} \\
& + (u_{\neg x_{n-1}} - u_{\neg x_n})(1 - \lambda_{\neg x_n}) \\
& + (u_{\neg x_{n-2}} - u_{\neg x_{n-1}})\big(1 - \max(\lambda_{\neg x_n}, \lambda_{\neg x_{n-1}})\big) \\
& + \cdots \\
& + (u_{\neg x_2} - u_{\neg x_3})\big(1 - \max(\lambda_{\neg x_n}, \ldots, \lambda_{\neg x_3})\big) \\
& + (u_{x_1} - u_{\neg x_2})\big(1 - \max(\lambda_{\neg x_n}, \ldots, \lambda_{\neg x_2})\big) \\
& + (u_{\neg x_1} - u_{x_1})\big(1 - \max(\lambda_{\neg x_n}, \ldots, \lambda_{\neg x_2}, \lambda_{x_1})\big) \\
& + (u_{\top} - u_{\neg x_1})\big(1 - \max(\lambda_{\neg x_n}, \ldots, \lambda_{\neg x_2}, \lambda_{x_1}, \lambda_{\neg x_1})\big) \\
= {} & 2 \\
& + 2(1 - \lambda_{\neg x_n}) \\
& + 2(1 - \lambda_{\neg x_{n-1}}) \\
& + \cdots \\
& + 2(1 - \lambda_{\neg x_3}) \\
& + (1 - \lambda_{\neg x_2}) \\
& + (1 - \lambda_{x_1}) \\
& + (1 - \lambda_{\neg x_1}) \\
= {} & 2 \\
& + 2(1 - \lambda_{\neg x_n}) \\
& + 2(1 - \lambda_{\neg x_{n-1}}) \\
& + \cdots \\
& + (1 - \lambda_{\neg x_2}) \\
& + (1 - \lambda_{\neg x_2}) \quad (\text{since by definition } \lambda_{x_1} = \lambda_{\neg x_2}) \\
& + (1 - \lambda_{\neg x_1}) \\
= {} & 2(1 - \lambda_{\neg x_n}) \\
& + 2(1 - \lambda_{\neg x_{n-1}}) \\
& + \cdots \\
& + 2(1 - \lambda_{\neg x_2}) \\
& + (1 - \lambda_{\neg x_1}) \\
= {} & 2 + 2(n - 1) - 2(\lambda_{\neg x_n} + \cdots + \lambda_{\neg x_2}) + 1 - \lambda_{\neg x_1} \\
= {} & 2n + 1 - 2(\lambda_{\neg x_n} + \cdots + \lambda_{\neg x_2}) - \lambda_{\neg x_1}.
\end{aligned}
$$

Table 6
Quantification of the decision tree in Fig. 4.

	x_1	$\neg x_1$	x_2	$\neg x_2$	x_3	$\neg x_3$
u	5	6	3	4	1	2
λ	0.49	0.7	0.343	0.49	0.24	0.343

It follows that:

$$\begin{aligned} Ch_N\big(L_{NC}^{\downarrow}\big) - Ch_N\big(L_C^{\uparrow}\big) &= 2n+1-2(\lambda_{x_n},\dots,\lambda_{x_1}) \\ &= -2n-1+2(\lambda_{\neg x_n},\dots,\lambda_{\neg x_2})+\lambda_{\neg x_1} \\ &= 2(\lambda_{\neg x_n},\dots,\lambda_{\neg x_2})+\lambda_{\neg x_1}-2(\lambda_{x_n},\dots,\lambda_{x_1}) \\ &= \lambda_{\neg x_1}-2\lambda_{x_n} \quad (\text{since by definition } \lambda_{\neg x_i}=\lambda_{x_{i-1}}). \end{aligned}$$

Recall that by construction $\lambda_{\neg x_1}=\epsilon$ and $\lambda_{x_n}=\epsilon^{n+1}$:

$Ch_N(L_{NC}^{\downarrow}) - Ch_N(L_C^{\uparrow})$ is equal to $\epsilon - 2\epsilon^{n+1}$. Since we have chosen ϵ in $[0,1]$ is such a way that $\epsilon^n < 0.5$, we get $Ch_N(L_{NC}^{\downarrow}) - Ch_N(L_C^{\uparrow}) > 0$, i.e. $Ch_N(L_{NC}^{\downarrow}) > Ch_N(L_C^{\uparrow})$.

This shows that the Choquet value of any non-contradictory strategy, if such a strategy exists, is greater than the Choquet value of the best contradictory strategy. Moreover, the CNF is consistent iff there exists a non-contradictory strategy. This means that the CNF is consistent iff there exists a strategy with a Choquet value Ch_N greater than $Ch_N(L_C^{\uparrow}) = 2n+1-2(\lambda_{\neg x_n}+\cdots+\lambda_{\neg x_2})-\lambda_{\neg x_1}$.

To illustrate the necessity-based transformation, we will consider the 3CNF $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ corresponding to the decision tree of Fig. 4. Table 6 represents necessary values for the quantification of this decision tree by following steps of transformation for $\epsilon = 0.8$. The best contradictory lottery $L_C^{\uparrow} = \langle 0.343/2, 0.49/4, 0, 49/5, 0.7/6, 1/7\rangle$ s.t. $Ch_N(L_C^{\uparrow}) = 2*3+1-(2*(0.343+0.49))-0.7 = 4.643$. If we consider the strategy $\sigma = \{(D_0 = d_0), (D_{Cl_1} = d_{Cl_1}^{x_1}), (D_{Cl_2} = d_{Cl_2}^{\neg x_3}), (D_{x_1} = d_{x_1}), (D_{x_2} = d_{x_2}), (D_{x_3} = d_{\neg x_3})\}$, the compound lottery corresponding to σ is $L = \langle 0.343/2, 0.343/3, 0.49/5, 1/7\rangle$ with $Ch_N(L) = 6.011 > Ch_N(L_C^{\uparrow}) = 4.643$ which means that the 3CNF is satisfiable by the instance $x_1 \wedge x_2 \wedge \neg x_3$.

NP-hardness of DT-OPT-Ch_Π

The hardness of DT-OPT-Ch_Π is proved by the reduction from a 3-SAT problem. More precisely, given a CNF $Cl = \{Cl_1,\dots,Cl_m\}$ on $X = \{X_1,\dots,X_n\}$ the possibility-based transformation into a possibilistic decision tree $\Pi\mathcal{T}$ can be ensured by the following procedure.

```
1. Fix $\epsilon$ such that $0 < \epsilon < 1$
2. Create a decision node $D_0$ as the root of $\Pi\mathcal{T}$
3. Create a chance node $H$
4. Add $H$ as a child of $D_0$
5. $\forall x_i \in X$
   • Create a decision node $D_{X_i}$ with two children $C_{x_i}$ and $C_{\neg x_i}$ s.t.:
   • $C_{x_i}$ is initialized with the lottery $\langle 1/0\rangle$
   • $C_{\neg x_i}$ is initialized with the lottery $\langle 1/0\rangle$
6. $\forall Cl_i \in Cl$, $\forall$ literal $l_j \in Cl_i$:
   Add to $C_{l_j}$ the lottery $\langle \epsilon^i / \sum_{k=0}^{i-1} 10^k\rangle$
```

Consider a 3CNF with m clauses $Cl = \{Cl_1,\dots,Cl_m\}$ on the set $X = \{X_1,\dots,X_n\}$ of propositional variables. The set of all literals is denoted by $\mathcal{L}$ and l_i denotes any literal in $\mathcal{L}$. In the following, we will use a constant $0 < \epsilon < 1$. Obviously, $\forall i \in \{1..m\}$, $\epsilon^i < 1$, and $i < j$ implies $\epsilon^i > \epsilon^j$.

A decision tree is built with a root node D_0 having as unique child a chance node (denoted by H) that branches on n decision nodes D_{X_i}, $i = 1\dots n$ (with a possibility degree equal to 1 for each child). Each D_{X_i} must make a decision on the value of X_i (i.e. x_i or $\neg x_i$), that is why it has two children, C_{x_i} and $C_{\neg x_i}$, which are chance nodes. Consider any literal $l_i \in \mathcal{L}$ and its corresponding chance node C_{l_i}, for the purpose of normalization of its possibility distribution, we systematically add to C_{l_i} a leave labeled with utility 0 with a possibility degree equal to 1. In addition, for any $Cl_i \in Cl$ satisfied by l_i, a leave node is added as a child of C_{l_i}, with a possibility degree equal to ϵ^i and a utility degree equal to $\sum_{k=0}^{i-1} 10^k$. In other words, the associated utility to Cl_1 (resp. $Cl_2, Cl_3, \dots, Cl_m$) is 1 (resp. $11, 111, \dots, \underbrace{1..1}_{m \text{ terms}}$)). One can check that:

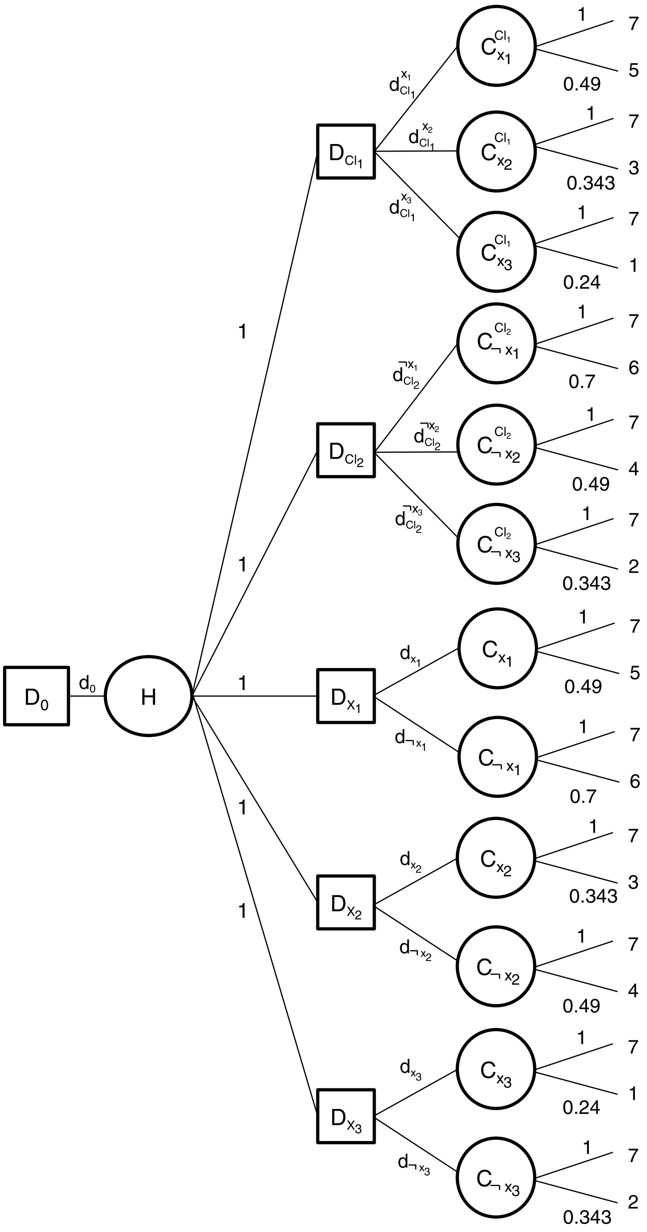

Fig. 4. NP-hardness of DT-OPT-Ch_N: transformation of the 3CNF $((x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3))$ s.t. $Cl_1 = (x_1 \vee x_2 \vee x_3)$ and $Cl_2 = (\neg x_1 \vee \neg x_2 \vee \neg x_3)$ with $\epsilon = 0.8$.

- The reduction is performed in $O(n+m)$.
- There is a bijection between the interpretation of the CNF and the admissible strategies.
- Ch_Π value of a strategy δ is equal to $\sum_{i=1,m,\delta \text{ satisfies } Cl_i} 10^{i-1} * \epsilon^i$.

The greater the number of clauses satisfied by a strategy, the greater its Ch_Π value. Hence the CNF is consistent iff there exists a strategy with a Choquet value (greater or) equal to $\sum_{i=1..m} 10^{i-1} * \epsilon^i$, i.e. which satisfies all clauses. Fig. 5 gives an example of this transformation for the 3CNF $((x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3))$ where $Cl_1 = (x_1 \vee x_2 \vee x_3)$ and $Cl_2 = (\neg x_1 \vee \neg x_2 \vee \neg x_3)$. With $\epsilon = 0.2$ we obtain $u(Cl_1) = 1$, $u(Cl_2) = 11$, $\pi(Cl_1) = 0.2$ and $\pi(Cl_2) = 0.04$. We have $\sum_{i=1,2} 10^{i-1} * 0.2^i = 0.64$. If we consider the strategy $\sigma = \{(D_0 = d_0), (D_{x_1} = d_{x_1}), (D_{x_2} = d_{x_2}), (D_{x_3} = d_{\neg x_3})$, the compound lottery corresponding to σ is $L = \langle 1/0, 0.2/1, 0.04/11\rangle$ with $Ch_\Pi(L) = 0.64$ which means that the 3CNF is satisfiable by the instance $x_1 \wedge x_2 \wedge \neg x_3$. □

Proof of Proposition 15. Let $U = \{u_1, u_2\}$ be the set of considered utilities.

First of all, let us remark that the composition of several lotteries built on U only leads by reduction to a simple lottery built on U only.

Let $L = \langle \lambda_1/u_1, \lambda_2/u_2\rangle$, $L' = \langle \lambda'_1/u_1, \lambda'_2/u_2\rangle$ and $L'' = \langle \lambda''_1/u_1, \lambda''_2/u_2\rangle$ be three simple lotteries built on U.

Let $L_1 = Reduction(\langle \alpha/L, \beta/L''\rangle)$ and $L_2 = Reduction(\langle \alpha/L', \beta/L''\rangle)$.

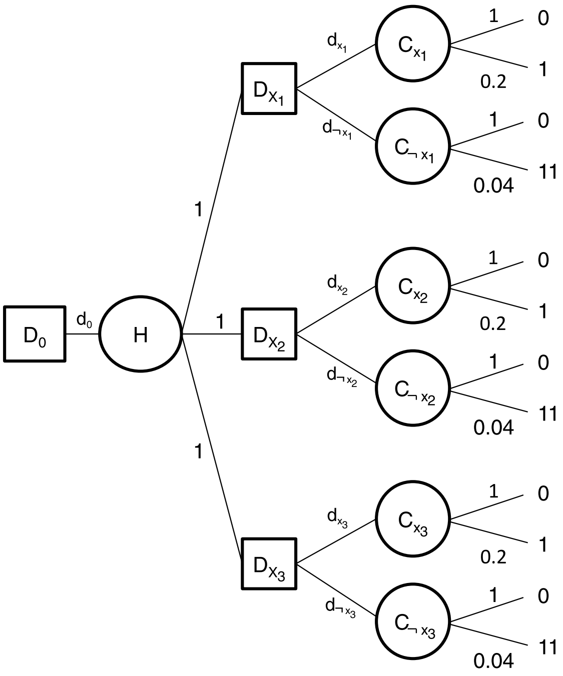

Fig. 5. NP-hardness of DT-OPT-Ch_Π: transformation of the 3CNF $((x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3))$ with $\epsilon = 0.2$.

Let us first show that the monotonicity property holds for Ch_N in binary decision trees.
Suppose that $Ch_N(L) \geqslant Ch_N(L')$; By definition we have:

$$Ch_N(L) = u_1 + (u_2 - u_1) * (1 - \lambda_1) \quad \text{and} \quad Ch_N(L') = u_1 + (u_2 - u_1) * (1 - \lambda'_1).$$

Thus $Ch_N(L) \geqslant Ch_N(L')$ implies $\lambda'_1 \geqslant \lambda_1$.

- If $\alpha = 1$: $L_1 = \langle \max(\lambda_1, \otimes(\beta, \lambda''_1))/u_1, \max(\lambda_2, \otimes(\beta, \lambda''_2))/u_2\rangle$ and $L_2 = \langle \max(\lambda'_1, \otimes(\beta, \lambda''_1))/u_1, \max(\lambda'_2, \otimes(\beta, \lambda''_2))/u_2\rangle$. Thus:

$$Ch_N(L_1) = u_1 + (u_2 - u_1) * \bigl(1 - \max\bigl(\lambda_1, \otimes(\beta, \lambda''_1)\bigr)\bigr) \quad \text{and}$$
$$Ch_N(L_2) = u_1 + (u_2 - u_1) * \bigl(1 - \max\bigl(\lambda'_1, \otimes(\beta, \lambda''_1)\bigr)\bigr).$$

  For both $\otimes = \min$ and $\otimes = *$, $\lambda'_1 \geqslant \lambda_1$ implies:

$$\max\bigl(\lambda'_1, \otimes\bigl(\beta, \lambda''_1\bigr)\bigr) \geqslant \max\bigl(\lambda_1, \otimes\bigl(\beta, \lambda''_1\bigr)\bigr).$$

  Thus $Ch_N(L_1) \geqslant Ch_N(L_2)$.
- If $\alpha < 1, \beta = 1$: $L_1 = \langle \max(\otimes(\alpha, \lambda_1), \lambda''_1)/u_1, \max(\otimes(\alpha, \lambda_2), \lambda''_2)/u_2\rangle$ and $L_2 = \langle \max(\alpha\lambda'_1, \lambda''_1)/u_1, \max(\alpha\lambda'_2, \lambda''_2)/u_2\rangle$. Thus:

$$Ch_N(L_1) = u_1 + (u_2 - u_1) * \bigl(1 - \max\bigl(\otimes(\alpha, \lambda_1), \lambda''_1\bigr)\bigr) \quad \text{and}$$
$$Ch_N(L_2) = u_1 + (u_2 - u_1) * \bigl(1 - \max\bigl(\otimes\bigl(\alpha, \lambda'_1\bigr), \lambda''_1\bigr)\bigr).$$

  For both $\otimes = \min$ and $\otimes = *$, $\lambda'_1 \geqslant \lambda_1$ implies: $\max(\otimes(\alpha, \lambda'_1), \lambda''_1) \geqslant \max(\otimes(\alpha, \lambda_1), \lambda''_1)$. Thus $Ch_N(L_1) \geqslant Ch_N(L_2)$.

Thus $Ch_N(L) \geqslant Ch_N(L')$ implies $Ch_N(L_1) \geqslant Ch_N(L_2)$: the monotonicity property is satisfied for binary decision trees of two levels. Since L_1 and L_2 are themselves binary, it is satisfied in any binary decision tree.

Let us now show that the monotonicity property holds for Ch_Π in binary decision trees. Thus, suppose that $Ch_\Pi(L) \geqslant Ch_\Pi(L')$;

$$Ch_\Pi(L) = u_1 + (u_2 - u_1)\lambda_2 \quad \text{and} \quad Ch_\Pi\bigl(L'\bigr) = u_1 + (u_2 - u_1)\lambda'_2.$$

Thus $Ch_\Pi(L) \geqslant Ch_\Pi(L')$ implies $\lambda_2 \geqslant \lambda'_2$.

- If $\alpha = 1$: $L_1 = \langle \max(\lambda_1, \otimes(\beta, \lambda''_1))/u_1, \max(\lambda_2, \otimes(\beta, \lambda''_2))/u_2\rangle$ and $L_2 = \langle \max(\lambda'_1, \otimes(\beta, \lambda''_1))/u_1, \max(\lambda'_2, \otimes(\beta, \lambda''_2))/u_2\rangle$. Thus:

$$Ch_\Pi(L_1) = u_1 + (u_2 - u_1) * \max\bigl(\lambda_2, \otimes(\beta, \lambda''_2)\bigr) \quad \text{and}$$
$$Ch_\Pi(L_2) = u_1 + (u_2 - u_1) * \max\bigl(\lambda'_2, \otimes(\beta, \lambda''_2)\bigr).$$

From $\lambda_2 > \lambda'_2$ it can be easily derived that $Ch_\Pi(L_1) \geqslant Ch_\Pi(L_2)$.

- If $\alpha < 1, \beta = 1$: $L_1 = \langle \max(\otimes(\alpha, \lambda_1), \lambda''_1)/u_1, \max(\otimes(\alpha, \lambda_2), \lambda''_2)/u_2\rangle$ and $L_2 = \langle \max(\alpha\lambda'_1, \lambda''_1)/u_1, \max(\alpha\lambda'_2, \lambda''_2)/u_2\rangle$. Thus:

$$Ch_\Pi(L_1) = u_1 + (u_2 - u_1) * \max\bigl(\otimes(\alpha, \lambda_2), \lambda''_2\bigr) \quad \text{and}$$
$$Ch_\Pi(L_2) = u_1 + (u_2 - u_1) * \max\bigl(\otimes\bigl(\alpha, \lambda'_2\bigr), \lambda''_2\bigr).$$

$Ch_\Pi(L_1) \geqslant Ch_\Pi(L_2)$ is thus easily derived from $\lambda_2 > \lambda'_2$. □

Proof of Proposition 16. Proposition 16 trivially follows from the definition of the Choquet integral: $Ch_\mu(L) = \sum_{i=1,m}(u_i - u_{i-1}).\mu(L \geqslant u_i)$: if $\mu(L' \geqslant u_i) \geqslant \mu(L'' \geqslant u_i)$ (i.e. if L' overcomes L'' according to μ) then each term of the sum is greater for L' than for L''; thus $Ch_\mu(L') \geqslant Ch_\mu(L'')$. □

Proof of Lemma 1. Obviously, $G^c_{Rev(G^c)}(u_1) = 1 = G^c(u_1)$.

Note that $\forall i = 2, n$, $Rev(G^c)(u_i) \geqslant Rev(G^c)(u_{i-1})$.

Hence $G^c_{Rev(G^c)}(u_i) = 1 - \max_{j=1,i-1} Rev(G^c)(u_j) = 1 - Rev(G^c)(u_{i-1})$.

Since $Rev(G^c)u_{i-1} = 1 - G^c(u_i)$, we get $G^c_{Rev(G^c)}(u_i) = G^c(u_i)$.

Thus $G^c_{Rev(G^c)} = G^c$. □

Proof of Proposition 17. From Lemma 1, we have $G^c_{Rev(G^c_{\mathcal{G}})}(u) = G^c_{\mathcal{G}}(u)$. By definition, $G^c_{\mathcal{G}}(u)$ is an upper envelop of $\mathcal{G}$. So, $Rev(G^c_{\mathcal{G}})$, which is equal to $G^c_{\mathcal{G}}$, overcomes any lottery L_i, $i = 1, \ldots, k$. □

References

[1] G. Choquet, Théorie des capacités, Tech. rep., Ann. Inst. Fourier (1953).

[2] D. Dubois, H. Fargier, P. Perny, Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach, Artif. Intell. 148 (1–2) (2003) 219–260.

[3] D. Dubois, H. Fargier, H. Prade, P. Perny, Qualitative decision theory: from Savage's axioms to nonmonotonic reasoning, J. ACM 49 (2002) 455–495.

[4] D. Dubois, L. Godo, H. Prade, A. Zapico, Making decision in a qualitative setting: from decision under uncertainty to case-based decision, in: 6th International Conference on Principles of Knowledge Representation and Reasoning (KR'1998), 1998, pp. 594–605.

[5] D. Dubois, H. Prade, Possibility Theory, an Approach to Computerized Processing of Uncertainty, Plenum Press, New York, NY, 1988.

[6] D. Dubois, H. Prade, Epistemic entrenchment and possibilistic logic, Artif. Intell. 50 (2) (1991) 223–239.

[7] D. Dubois, H. Prade, Possibility theory as a basis for qualitative decision theory, in: International Joint Conference on Artificial Intelligence (IJCAI'95), 1995, pp. 1924–1930.

[8] H. Fargier, N. Ben Amor, W. Guezguez, On the complexity of decision making in possibilistic decision trees, in: Uncertainty in Artificial Intelligence (UAI'11), 2011, pp. 203–210.

[9] H. Fargier, P. Perny, Qualitative models for decision under uncertainty without the commensurability assumption, in: Uncertainty in Artificial Intelligence (UAI'99), 1999, pp. 188–195.

[10] L. Garcia, R. Sabbadin, Possibilistic influence diagrams, in: European Conference on Artificial Intelligence (ECAI'06), 2006, pp. 372–376.

[11] P. Giang, P.P. Shenoy, A comparison of axiomatic approaches to qualitative decision making using possibility theory, in: Uncertainty in Artificial Intelligence (UAI'01), 2001, pp. 162–170.

[12] P. Giang, P.P. Shenoy, A qualitative linear utility theory for spohn's theory of epistemic beliefs, in: Uncertainty in Artificial Intelligence (UAI'2000), 2000, pp. 220–229.

[13] P. Giang, P.P. Shenoy, Two axiomatic approaches to decision making using possibility theory, Eur. J. Oper. Res. 162 (2) (2005) 450–467.

[14] W. Guezguez, N. Ben Amor, K. Mellouli, Qualitative possibilistic influence diagrams based on qualitative possibilistic utilities, Eur. J. Oper. Res. 195 (1) (2009) 223–238.

[15] G. Jeantet, O. Spanjaard, Rank-dependent probability weighting in sequential decision problems under uncertainty, in: International Conference on Automated Planning and Scheduling (ICAPS'08), 2008, pp. 148–155.

[16] G. Jeantet, O. Spanjaard, Computing rank dependent utility in graphical models for sequential decision problems, Artif. Intell. 175 (7–8) (2011) 1366–1389.

[17] D. Kikuti, F.G. Cozman, C.P. de Campos, Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities, in: IJCAI'05 Workshop on Advances in Preference Handling, Edinburgh, United Kingdom, 2005, pp. 118–123.

[18] N. Huntley, M.C.M. Troffae, Normal form backward induction for decision trees with coherent lower previsions, Ann. Oper. Res. 195 (1) (2012) 111–134.

[19] J. Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1948.

[20] J. Pearl, From conditional oughts to qualitative decision theory, in: Uncertainty in Artificial Intelligence (UAI'93), 1993, pp. 12–20.

[21] P. Perny, O. Spanjaard, P. Weng, Algebraic Markov decision processes, in: International Joint Conference on Artificial Intelligence (IJCAI'05), 2005, pp. 1372–1377.

[22] J. Quiggin, A theory of anticipated utility, J. Econ. Behav. Organ. 3 (1982) 323–343.

[23] H. Raifaa, Decision Analysis: Introductory Lectures on Choices under Uncertainty, Addison-Wesley, Reading, MA, 1968.

[24] Y. Rebille, Decision making over necessity measures through the Choquet integral criterion, Fuzzy Sets Syst. 157 (23) (2006) 3025–3039.

[25] R. Sabbadin, A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments, in: Uncertainty in Artificial Intelligence (UAI'99), 1999, pp. 567–574.

[26] R. Sabbadin, H. Fargier, J. Lang, Towards qualitative approaches to multi-stage decision making, Int. J. Approx. Reason. 19 (3–4) (1998) 441–471.

[27] D. Schmeidler, Subjective probability and expected utility without additivity, Econometrica 57 (1989) 517–587.

[28] W. Spohn, A general non-probabilistic theory of inductive reasoning, in: Uncertainty in Artificial Intelligence (UAI'90), 1990, pp. 315–322.
[29] N. Wilson, An order of magnitude calculus, in: Uncertainty in Artificial Intelligence (UAI'95), 1995, pp. 548–555.
[30] M. Yaari, The dual theory of choice under risk, Econometrica 55 (1987) 95–115.
[31] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets Syst. 1 (1978) 3–28.