# Complexity of Solving Decision Trees with Skew-Symmetric Bilinear Utility

**Hugo Gilbert**
Sorbonne Universités, UPMC Univ Paris 06
CNRS, LIP6 UMR 7606
4 place Jussieu
75005 Paris, France

**Olivier Spanjaard**
Sorbonne Universités, UPMC Univ Paris 06
CNRS, LIP6 UMR 7606
4 place Jussieu
75005 Paris, France

## Abstract

We study the complexity of solving decision trees with a Skew-Symmetric Bilinear (SSB) utility function. The SSB model is an extension of Expected Utility (EU) with enhanced descriptive possibilities. Unlike EU, the optimality principle does not hold for SSB, which makes its optimization trickier. We show that determining an SSB optimal plan is NP-hard if one only considers deterministic plans while it is polynomial time if one allows randomized plans. With the Weighted EU model (a special case of SSB), the problem becomes polynomial in both settings. Our numerical tests show the operationality of the methods.

## 1 INTRODUCTION

This paper deals with planning under risk. In this setting, an agent repeatedly chooses actions to perform in order to maximize a given decision criterion. A solution to a planning problem is a plan, *i.e.* a sequence of actions conditioned by events. The difficulty of finding the optimal plan relies in the fact that actions' consequences are uncertain. Thus, a plan results in a probability distribution over possible outcomes (*i.e.* lotteries) and comparing plans amounts to comparing their respective lotteries. A popular decision criterion to compare lotteries is the Expected Utility (EU) model (von Neumann and Morgenstern, 1947). In this model, preferences over the set $\mathcal{X}$ of possible outcomes are modeled using a utility function which assigns a numerical value to each element $x \in \mathcal{X}$. Then, the preferences are lifted from outcomes to lotteries using expectation. More formally, a lottery $p$ is preferred to a lottery $q$, denoted by $p \succ q$, iff $u(p) > u(q)$, with $u(p) = \sum_{x\in\mathcal{X}} p(x)u(x)$[1]. However, despite its appeal from a *normative* viewpoint, the EU model does not make it possible to describe all observed preferences (*descriptive* viewpoint). We present here two examples of observed preferences that EU cannot accommodate while they seem rational.

[1]with $p(x)$ the probability that $p$ yields outcome $x$.

**Example 1** (Gardner's dice)**.** *Consider three dice (Gardner, 1970): die $A$ with faces $(1,4,4,4,4,4)$, die $B$ with faces $(3,3,3,3,3,6)$ and die $C$ with faces $(2,2,2,5,5,5)$. Consider a very simple two-player game where each player throws a die, and the player which throws the highest value wins. The probability to win with $A$ (resp. $B$, $C$) against $B$ (resp. $C$, $A$) is $25/36$ (resp. $21/36$, $21/36$). Thus, there is a "winning cycle": $A$ wins against $B$, $B$ wins against $C$ and $C$ wins against $A$. In this situation, it is therefore not possible to represent the preferences over dice by using an EU criterion (i.e., assigning a utility to each face in order to evaluate each die by the expected utility of the rolled face) because EU is not able to account for* intransitive *preferences.*

**Example 2** (Allais' paradox)**.** *We present in Table 1 a very simple variant of this well-known paradox (variant by Kahneman and Tversky, 1979). Each row corresponds to a lottery $p$, $q$, $p'$ or $q'$ and each cell indicates the probability to win* 0\$, 3000\$ *or* 4000\$. *Allais' paradox stipulates that most people prefer lottery $p$ to lottery $q$ but prefer lottery $q'$ to lottery $p'$. However, lottery $p'$ (resp. $q'$) is simply the mixture of lottery $p$ (resp. $q$), with probability* 0.25*, and a sure amount of* 0\$, *with probability* 0.75. *Thus, those preferences violate the* independence *axiom which holds in EU theory. In short, the independence axiom states that $p \succ q \Rightarrow \alpha p + (1-\alpha)r \succ \alpha q + (1-\alpha)r$ where $r$ is any lottery, $\alpha \in (0,1)$ and $\alpha p + (1-\alpha)r$ denotes the mixture where $p$ (resp. $r$) is obtained with probability $\alpha$ (resp. $1-\alpha$). This makes it impossible to account for such preferences with the EU model.*

In this paper, we study two models able to cope with the preferences observed in the above examples: the Skew-Symmetric Bilinear (SSB) utility model (Fishburn, 1984b, 1982) is able to account for intransitive and non-

| $x$ | 0$ | 3000$ | 4000$ |
|---|---|---|---|
| $p(x)$ | 0 | 1 | 0 |
| $q(x)$ | 0.2 | 0 | 0.8 |
| $p'(x)$ | 0.75 | 0.25 | 0 |
| $q'(x)$ | 0.8 | 0 | 0.2 |

Table 1: Allais' paradox.

independent preferences, while the Weighted Expected Utility (WEU) model (Chew, 1983; Fishburn, 1983) copes with non-independent preferences. Note that the WEU model is a subclass of the SSB model. Our aim is to identify the complexity of determining an optimal plan according to each of these two models in a sequential decision problem under risk.

A standard way of modeling a sequential decision problem under risk is to use a *decision tree* representing all decision steps and possible events. In a decision tree, the number of plans exponentially increases with the number of decision steps, thus the problem of choosing a plan is of combinatorial nature. The choice of a plan can be made in several manners according to the behavior of the Decision Maker (DM). We now briefly describe the most studied types of behaviors. A *consequentialist* DM selects a plan looking only at the possible futures (regardless of the past or counterfactual history); among consequentialist DMs, the *sophisticated* ones select a plan starting from the anticipated future decisions and rolling back to the present (in a recursive manner); on the contrary, a *resolute* DM considers all possible plans from the present time, chooses one and commits to it without deviating thereafter. The behavior of a resolute DM is therefore non-consequentialist because her choices in the future will depend on past and counterfactual events.

Note that an EU optimizer is both sophisticated and resolute. This is related to the fact that the Bellman optimality principle holds for the EU criterion (i.e., a subplan of an optimal plan is optimal). However, it is not the case for criteria able to account for intransitive and/or non-independent preferences because, contrary to the EU model, they violate the Bellman optimality principle. A DM whose behavior is consistent with the SSB/WEU model is thus either sophisticated or resolute. For a sophisticated DM, determining an optimal plan for the SSB/WEU model can be performed with a usual rolling back procedure. The drawback is the possibility to act as a *money pump* (Tversky, 1969). As in the Gardner's dice example, assume a DM prefers $A$ to $B$, $B$ to $C$, and $C$ to $A$. It is reasonable to assume that she is willing to pay a small amount $\varepsilon$ of money to replace $A$ (resp. $C$, $B$) by $C$ (resp. $B$, $A$). Consider now a sequential decision problem where the DM initially receives die $A$ for free, then is proposed to switch for $C$, then to switch for $B$, then to switch for $A$, where each exchange has a cost of $\varepsilon$. A sophisticated DM will make all exchanges, ending up with die $A$ as initially, but having spent an amount of $3\varepsilon$ (for a sufficiently small $\varepsilon$). Furthermore, even if the preferences are transitive, it is known that a sophisticated DM with non-independent preferences may end up with a stochastically dominated plan (Hammond, 1988).

|  | SSB | WEU |
|---|---|---|
| *deterministic* | NP-hard | P |
| *randomized* | P | P |

Table 2: Synthesis of the contributions.

Interestingly, a resolute DM consistent with SSB is subject to none of these drawbacks. For a resolute DM, there are several manners to define an optimal strategy:
– *Dictatorship of the root*: the "root" refers to the situation at the start of the decision process; with this criterion, the DM chooses a plan that maximizes the SSB/WEU score at the root. The use of this criterion has been advocated by McClennen (1990).
– *Resolute choice with selves*: it consists in considering the present decision situation as well as each possible future decision situations as a self who represents the DM at the time and state when the decision is made (Jaffray, 1998); one then aims at determining a plan achieving a compromise between the different selves of the DM, i.e. a plan that remains suitable for all selves.
In this paper, we study the former approach (dictatorship of the root). We distinguish between two settings regarding the set of feasible plans. In the *deterministic* setting (most prominent one in the literature), one focuses on plans where a decision is deterministically selected in each possible decision situation. In the *randomized* setting, one considers the enlarged set of plans where a decision is randomly selected (according to a predefined probability distribution) in each possible situation.

After proving that, in the deterministic setting, "optimizing" (in a sense to be formalized later) an SSB utility function in a decision tree is NP-hard, we will see that, interestingly, optimizing a WEU function in a decision tree can be done in polynomial time. To the best of our knowledge, WEU is the first class of criteria able to cope with Allais' paradox and which does not lead to an NP-hard problem for determining an optimal plan. Table 2 summarizes our contributions.

## 2 RELATED WORKS

Several non-EU models have been investigated in sequential decision making under uncertainty.

**Rank-Dependent Utility (RDU) model.** The RDU model proposed by Quiggin (1993), where one specifies a probability distortion function, is compatible with Allais' paradox. Nielsen and Jaffray (2006) have studied the solution of sequential decision problems with RDU, under the resolute choice with selves paradigm. They proposed an operational approach eliminating any plan that appears to be largely suboptimal for RDU in some decision situation. Their approach returns an RDU-optimal plan (viewed from the root) among the remaining ones, such that no other plan stochastically dominates it. Jeantet et al. (2012) proposed another implementation of RDU theory in decision trees. The difference with the previous approach is that the cooperation between selves is formalized through the use of a weighted max regret criterion, where regrets measure RDU losses. Lastly, Jeantet and Spanjaard (2008) investigated the solution of sequential decision problems with the RDU model, under the dictatorship of the root paradigm. They showed that the problem is NP-hard and designed a branch and bound procedure to solve it.

**Imprecise Probabilities.** When probabilities are imprecise (i.e., a set of probability measures should be taken into account in the decisions instead of a single measure), several decision criteria can be used to evaluate a plan. In this setting, a pessimistic DM will make the decision that maximizes the worst possible expected utility. This is known as the Γ-maximin criterion. Conversely, an optimistic agent will make the decision that maximizes the best possible expected utility. This is known as the Γ-maximax criterion. Kikuti et al. (2011) proposed algorithms enabling a sophisticated DM to determine her preferred plan w.r.t. both criteria. Their algorithms rely on linear/multilinear programming. Subsequently, Fargier et al. (2011) studied the optimization of the Γ-maximin criterion under the dictatorship of the root paradigm. They showed that even the evaluation of a plan according to the Γ-maximin criterion is NP-hard, and proposed a procedure to determine an optimal plan derived from preliminary results by Huntley and Troffaes (2008). The Hurwicz criterion for imprecise probabilities is a convex combination of the Γ-maximin and the Γ-maximax criteria that can model intermediate attitudes w.r.t. ambiguity (Jaffray and Jeleva, 2007). The optimization of the Hurwicz criterion has been studied under the dictatorship of the root paradigm (Jeantet and Spanjaard, 2009). Once again, the determination of an optimal plan for the Hurwicz criterion is NP-hard.

**Skew-Symmetric Bilinear utility.** The use of SSB utilities in finite horizon Markov decision processes has been studied by Gilbert et al. (2015) under the dictatorship of the root paradigm. The authors showed that an optimal randomized policy always exists, and designed a game-theoretic solution procedure. This work was latter extended to the setting of reinforcement learning (Gilbert et al., 2016) and was used in a document treatment chain application (Nicart et al., 2016). The complexity of determining an SSB optimal plan remains an open question, which we address in this paper for sequential decision problems represented by decision trees.

Before describing the formalism of decision trees, we present the SSB and WEU models in the next section.

## 3 SSB AND WEU MODELS

### 3.1 The SSB model

In the SSB model (Fishburn, 1984b, 1982), the preferences of the agent under certainty are represented via a binary functional $\varphi$ over pairs of outcomes $(x,y) \in \mathcal{X}^2$, with $x > y \Leftrightarrow \varphi(x,y) > 0$. The value $\varphi(x,y)$ measures the intensity with which the agent prefers outcome $x$ to $y$. Function $\varphi$ is assumed to be skew-symmetric and bilinear w.r.t. the mixture operation on lotteries. More formally, skew-symmetry means that $\varphi(x,y) = -\varphi(y,x)$, and bilinearity means that $\varphi(\sum_i \lambda_i p_i, q) = \sum_i \lambda_i \varphi(p_i, q)$ and $\varphi(p, \sum_i \lambda_i q_i) = \sum_i \lambda_i \varphi(p, q_i)$, where $\sum_i \lambda_i p_i$ is the lottery $p$ defined by $p(x) = \sum \lambda_i p_i(x)$. The preferences are then lifted to probability distributions by expectation: $p \succ q \Leftrightarrow \varphi(p,q) = \sum_{x,y \in \mathcal{X}^2} p(x)q(y)\varphi(x,y) > 0$. SSB utility theory has strong descriptive abilities. Not only can it account for Allais' paradox but it can also allow intransitive preferences which have been observed in various experiments.

**Example 3** (Gardner's dice cont'd). *Gardner's dice correspond to the following lotteries on $\mathcal{X} = \{1, \ldots, 6\}$:*

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $p_A$ | 1/6 | 0 | 0 | 5/6 | 0 | 0 |
| $p_B$ | 0 | 0 | 5/6 | 0 | 0 | 1/6 |
| $p_C$ | 0 | 1/2 | 0 | 0 | 1/2 | 0 |

*By setting $\varphi(x,y) = 1$ if $x > y$, and $\varphi(x,y) = -1$ if $x < y$, $\varphi(p,q)$ corresponds then to the probability that $p$ beats $q$ minus the probability that $q$ beats $p$. The obtained SSB utilities in the example are:*

$$\begin{aligned} \varphi(p_A,p_B) &= 25/36 - 11/36 &= 14/36, \\ \varphi(p_B,p_C) &= 21/36 - 15/36 &= 6/36, \\ \varphi(p_C,p_A) &= 21/36 - 15/36 &= 6/36. \end{aligned}$$

*Thus, $\varphi(p_A,p_B){>}0$, $\varphi(p_B,p_C){>}0$ and $\varphi(p_C,p_A){>}0$, which is consistent with the relation "more likely to win" between dice (i.e., $p_A{\succ}p_B{\succ}p_C{\succ}p_A$).*

As SSB utility can lead to intransitive preferences, the existence of an *optimal* lottery is not obvious. Given a set $\mathcal{L} = \{p_1, \ldots, p_n\}$ of lotteries, the SSB criterion induces a *weighted tournament* on $\mathcal{L}$, i.e., for each pair $p, q$ of lotteries such that $\varphi(p,q) > 0$, $\varphi(p,q)$ represents the intensity with which $p$ is preferred to $q$. Multiple rules exist for determining the winner(s) of a weighted tournament (Fischer et al., 2016). We adopt in this paper the minimax rule (Young, 1977), also known as Condorcet's rule or the Simpson-Kramer method, where each lottery $p$ is evaluated by the highest intensity with which another lottery is preferred to $p$, and one selects a lottery with minimal evaluation. More formally, one seeks a lottery $p$ in $\arg\min_{p\in\mathcal{L}}\max_{q\in\mathcal{L}}\varphi(q,p)$.

If one enlarges the set of possible lotteries to the convex hull $CH(\mathcal{L})$ of $\mathcal{L}$ where $CH(\mathcal{L}) = \{p : p=\sum_{i=1}^{n}\lambda_i p_i$ with $\sum_{i=1}^{n}\lambda_i=1$ and $\lambda_i\geq 0,\ \forall i\}$, Fishburn (1984a) showed that an optimal lottery $p\in CH(\mathcal{L})$ w.r.t. the minimax rule has the desirable property that $\varphi(p,q)\geq 0$ for all $q$. However, a DM may not accept to use a mixed lottery (i.e., a lottery in $CH(\mathcal{L})\setminus\mathcal{L}$). Thus, we study both optimization in $\mathcal{L}$ and in $CH(\mathcal{L})$.

The WEU criterion is a special case of SSB utilities, that enforces transitivity while still being able to cope with Allais' paradox. This is the topic of the next section.

### 3.2 The WEU model

The Weighted Expected Utility (WEU) theory, developed by Chew (1983), is obtained from the SSB utility theory by adding a transitivity axiom (Fishburn, 1983). The WEU criterion relies on two functions $u$ and $w$ (with $w > 0$) defined on $\mathcal{X}$ such that $\varphi(x,y) = u(x)w(y) - u(y)w(x)$. These two functions are lifted to lotteries by linearity in probabilities: $u(p) = \sum_{x\in\mathcal{X}} p(x)u(x)$ and $w(p) = \sum_{x\in\mathcal{X}} p(x)w(x)$. Consequently: $p \succ q \Leftrightarrow u(p)w(q)-u(q)w(p)>0$ which can be rewritten as:

$$p \succ q \Leftrightarrow v(p) = u(p)/w(p) > v(q) = u(q)/w(q) \quad (1)$$

In the sequel, we will continue to use $v$ to denote the ratio $u/w$. As WEU assigns a score $v(p)$ to each lottery $p$, it is unable to accommodate intransitive preferences. However, it can still accommodate Allais' paradox:

**Example 4** (Allais' paradox cont'd)**.** *We rescale the possible gains in Table 1 from interval* $[0,4000]$ *to interval* $[0,1]$ *(3000 becomes 3/4 and 4000 becomes 1). Consider the pair of functionals* $(u,w)$ *defined by* $u(x) = x^2$ *and* $w(x) = 1-\sqrt{x}+x^2$*, leading to the following values:*

| | $p$ | $q$ | $p'$ | $q'$ |
|---|---|---|---|---|
| $u$ | 0.5625 | 0.8 | 0.1406 | 0.2 |
| $w$ | 0.6965 | 1 | 0.9241 | 1 |

*We can easily check that the WEU model using these two functions is compatible with Allais' paradox:*

- $u(p)w(q) - u(q)w(p) \approx 0.005 > 0 \Rightarrow p \succ q$
- $u(q')w(p') - u(p')w(q') \approx 0.044 > 0 \Rightarrow q' \succ p'$

Before investigating the optimization of SSB and WEU utilities in sequential decision problems, we recall the formalism of decision trees in the next section.

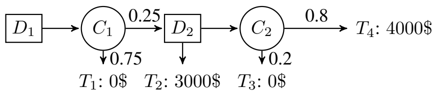

Figure 1: Allais' paradox as a decision tree problem.

## 4 DECISION TREE

A decision tree represents a sequential decision problem with three types of nodes: the *decision nodes* (represented by squares), the *chance nodes* (represented by circles), and the terminal nodes (the leaves of the tree). The branches starting from a decision node correspond to different possible decisions, while the ones starting from a chance node correspond to different possible events, the probabilities of which are known. The values indicated at the leaves correspond to the resulting outcomes. Such a decision tree is given in Figure 1. This decision tree is related to Allais' paradox. Indeed in node $D_2$, the agent needs to choose between lotteries $p$ and $q$ while in node $D_1$, the two possible plans yield lotteries $p'$ and $q'$, where $p$, $q$, $p'$ and $q'$ are as defined in Example 2.

More formally, a decision tree $\mathcal{T}$ is composed of a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$. The root node is denoted by $N_r$ and the set of decision nodes, chance nodes and terminal nodes are respectively denoted by $\mathcal{N}_D$, $\mathcal{N}_C$ and $\mathcal{N}_T$. We define $\mathcal{E}_D := \{(D,N)\in\mathcal{E} : D\in\mathcal{N}_D\}$. Every edge $E=(C,N)\in\mathcal{E}$ such that $C\in\mathcal{N}_C$ is weighted by probability $p_E$ of the corresponding event; every terminal node $T\in\mathcal{N}_T$ is labeled by the resulting outcome $o(T)$. Lastly, $\mathcal{S}(N)$ denotes the set of successors of $N$.

A plan in a decision tree is called a *strategy*. A strategy $\delta$ corresponds to a set $\mathcal{E}_\delta \subseteq \mathcal{E}_D$. The set of all feasible strategies is denoted by $\Delta$. Let $\chi^\delta := (\chi_E)_{E\in\mathcal{E}_D}$ denote the incidence vector of a strategy $\delta$ (i.e., $\chi_E = 1$ if $E\in\mathcal{E}_\delta$ and $\chi_E = 0$ otherwise). The set of feasible strategies can be characterized as the incidence vectors satisfying the following constraints:

$$\sum_{N\in\mathcal{S}(N_r)} \chi_{(N_r,N)} = 1$$

$$\forall D\in\mathcal{N}_D :$$
$$\text{if } \chi_{(D,C)}=0 \text{ then } \sum_{N\in\mathcal{S}(D')} \chi_{(D',N)} = 0 \quad \forall D'\in\mathcal{S}(C)\cap\mathcal{N}_D$$
$$\text{otherwise } \sum_{N\in\mathcal{S}(D')} \chi_{(D',N)} = 1 \quad \forall D'\in\mathcal{S}(C)\cap\mathcal{N}_D$$

For instance, in Figure 1, $\mathcal{E}_D$ includes the three edges $E_1=(D_1,C_1)$, $E_2=(D_2,C_2)$ and $E_3=(D_2,T_2)$, and there are two possible deterministic strategies characterized by incidence vectors $(\chi_{E_1}=1,\chi_{E_2}=1,\chi_{E_3}=0)$ and $(\chi_{E_1}=1,\chi_{E_2}=0,\chi_{E_3}=1)$.

By abuse of notation, we will denote a strategy indifferently by $\delta$ or $\chi^\delta$. Following the same convention, we will denote by $\Delta$ both the set of feasible strategies and the set of corresponding incidence vectors.

*Randomized strategies* are obtained by relaxing the domain of variables $\chi_E$ from $\{0,1\}$ to $[0,1]$ while keeping the same constraints. In the decision tree represented in Figure 1, an example of a randomized strategy is given by vector $(\chi_{E_1} = 1, \chi_{E_2} = 0.5, \chi_{E_3} = 0.5)$. This corresponds to the situation where the DM in decision node $D_2$ will choose with equal probabilities to go either in $T_2$ or in $C_2$. We denote by $\widehat{\Delta}$ the set of feasible randomized strategies (or the corresponding incidence vectors).

As usual in sequential decision under risk, a strategy $\delta$ induces a lottery $p_\delta$ over outcomes. Let $p_N^\delta$ denote the probability to reach node $N$ when following strategy $\delta$. Values $p_N^\delta$ can be recursively computed as follows:
$p_{N_r}^\delta = 1$
$\forall D \in \mathcal{N}_D, \forall N \in \mathcal{S}(D), p_N^\delta = p_D^\delta * \chi_{(D,N)}^\delta$
$\forall C \in \mathcal{N}_C, \forall N \in \mathcal{S}(C), p_N^\delta = p_C^\delta * p_{(C,N)}$
The lottery over outcomes $p_\delta$ induced by $\delta$ is defined by $p_\delta(x) = \sum_{T \in \mathcal{N}_T : o(T)=x} p_T^\delta$ for $x \in \mathcal{X}$.

Let $P = \{p_\delta : \delta \in \Delta\}$ denote the image of deterministic strategies in the space of lotteries, and $\widehat{P} = \{p_\delta : \delta \in \widehat{\Delta}\}$ denote the image of randomized strategies in the space of lotteries. It is worth noting that $CH(P) = \widehat{P}$, where $CH(P) = \{p_\delta : \delta \in CH(\Delta)\}$. This is due to Carathéodory's theorem. As a consequence of this observation, optimizing over $\widehat{P}$ is equivalent to optimizing over $CH(P)$, and therefore there always exists a strategy $\delta \in \widehat{\Delta}$ such that $\varphi(p_\delta, p_{\delta'}) \geq 0$ for all feasible strategies $\delta' \in \widehat{\Delta}$ (by Fishburn's result recalled in Section 3.1).

Note that the number of deterministic strategies can possibly grow exponentially with the size of the decision tree, i.e. the number of decision nodes (this number has indeed the same order of magnitude as the number of nodes in $\mathcal{T}$). Indeed, one easily shows that there are $\Theta(2^{\sqrt{|\mathcal{N}_\mathcal{D}|}})$ strategies in a complete binary decision tree $\mathcal{T}$. For this reason, even in the deterministic setting, it is necessary to develop an optimization algorithm to determine the optimal strategy. It is well known that the rolling back method makes it possible to compute in linear time an optimal strategy w.r.t. EU. Indeed such a strategy satisfies the optimality principle: any substrategy of an optimal strategy is itself optimal, where a substrategy is the restriction of a strategy in $\mathcal{T}$ to a subtree. Starting from the leaves, one computes recursively for each node the EU of an optimal substrategy: the optimal EU for a chance node equals the expectation of the optimal utilities of its successors; the optimal EU for a decision node equals the maximum EU of it's successors.

Unfortunately, note that any criterion able to cope with Allais' paradox necessarily violates the optimality principle. Indeed in the decision tree represented in Figure 1 such a criterion would have $\{(D_2, T_2)\}$ as optimal substrategy from node $D_2$. However, the optimal strategy from node $D_1$ is $\{(D_1, C_1), (D_2, C_2)\}$. For this reason, the optimization of such criteria in a decision tree is tricky. This is the topic of the following sections. The optimization problems we consider are:

$$(\mathcal{R}) : \ \min_{\delta \in \widehat{\Delta}} \max_{\delta' \in \widehat{\Delta}} \varphi(p_{\delta'}, p_\delta)$$
$$(\mathcal{D}) : \ \min_{\delta \in \Delta} \max_{\delta' \in \Delta} \varphi(p_{\delta'}, p_\delta)$$

Problem $(\mathcal{R})$ (resp. $(\mathcal{D})$) aims at determining an optimal randomized (resp. deterministic) strategy for the SSB criterion. If $\varphi(p,q) = w(q)u(p) - w(p)u(q)$, then the problem amounts to determine an optimal WEU strategy.

## 5 OPTIMIZING THE SSB CRITERION

We start by presenting a Linear Program (LP) involving a polynomial number of variables and constraints (in the size of the decision tree) for solving optimization problem $(\mathcal{R})$. Thus, by polynomial complexity of linear programming (Khachiyan, 1980), determining an SSB optimal randomized strategy in a decision tree is a polynomial time problem. To this end, we start by noting that:

$$\max_{\delta' \in \widehat{\Delta}} \varphi(p_{\delta'}, p_\delta) = \max_{\delta' \in \Delta} \varphi(p_{\delta'}, p_\delta)$$

for any $\delta \in \widehat{\Delta}$. Indeed, if $\Delta = \{\delta_1, \ldots, \delta_n\}$, then there exist $\lambda_i \geq 0$ summing up to 1 such that $p_{\delta'} = \sum_{i=1}^n \lambda_i p_{\delta_i}$ and we have the following sequence of relations:

$$\varphi(\sum_i \lambda_i p_{\delta_i}, p_\delta) = \sum_{i=1}^n \lambda_i \varphi(p_{\delta_i}, p_\delta) \leq \max_i \varphi(p_{\delta_i}, p_\delta).$$

We now rewrite the objective function in a way highlighting the fact that determining a strategy $\delta'$ that is most preferred to strategy $\delta$ can be done by rolling back the decision tree $\mathcal{T}$ with a particular utility function $V_\delta$.
By linearity of $\varphi$, we have:

$$\max_{\delta' \in \Delta} \varphi(p_{\delta'}, p_\delta) = \max_{\delta' \in \Delta} \sum_{x \in \mathcal{X}} p_{\delta'}(x) \varphi(x, p_\delta)$$

This latter expression can be rewritten as follows:

$$\max_{\delta' \in \Delta} \sum_{x \in \mathcal{X}} p_{\delta'}(x) \varphi(x, p_\delta) = \max_{\delta' \in \Delta} \sum_{x \in \mathcal{X}} p_{\delta'}(x) V_\delta(x)$$

where utilities $V_\delta(x)$ are defined by $V_\delta(x) = \varphi(x, p_\delta)$.

We now give the two classes of constraints involved in our LP to solve optimization problem $(\mathcal{R})$. The first class of constraints involves a set of variables $p_N$, describing all feasible randomized strategies $\delta \in \widehat{\Delta}$, where $p_N$ is the probability that node $N$ is reached with strategy $\delta$:

$$p_{N_r} = 1$$
$$p_D = \sum_{N \in \mathcal{S}(D)} p_N, \forall D \in \mathcal{N}_D$$
$$p_{(C,N)} p_C = p_N, \ \forall C \in \mathcal{N}_C, \ \forall N \in \mathcal{S}(C)$$
$$p_N \geq 0, \forall N \in \mathcal{N}$$

The strategy $\delta$ (more precisely, its incidence vector $\chi^{\delta}$) can be recovered from variables $p_N$ by using equation $\chi^{\delta}_{(D,N)} = p_N/p_D$ for each edge $(D,N) \in \mathcal{E}_D$. The second class of constraints aims at determining $\max_{\delta' \in \Delta} \sum_{x \in \mathcal{X}} p_{\delta'}(x) V_{\delta}(x)$ for a given strategy $\delta$. As indicated above, this value can be computed by rolling back the decision tree, which amounts to satisfy the following Bellman's equations, where $q_N$ denotes the EU value in node $N$ according to utility function $V_{\delta}$:

$$\begin{aligned} &q_T = V_{\delta}(o(T)), \forall T \in \mathcal{N}_T \\ &q_C = \sum_{N \in \mathcal{S}(C)} p_{(C,N)} q_N, \forall C \in \mathcal{N}_C \\ &q_D \geq q_N, \forall N \in \mathcal{S}(D),\ \forall D \in \mathcal{N}_D \\ &q_N \in \mathbb{R}, \forall N \in \mathcal{N} \end{aligned}$$

We have therefore $q_{N_r} = \max_{\delta' \in \Delta} \sum_{x \in \mathcal{X}} p_{\delta'}(x) V_{\delta}(x)$. Putting together the objective function $\min q_{N_r}$ and both classes of constraints, and replacing values $V_{\delta}(o(T))$ by $\sum_{T' \in \mathcal{N}_T} p_{T'} \varphi(o(T), o(T'))$, we obtain the final program $\mathcal{P}_{SSB}$ given below, that enables to determine an SSB optimal randomized strategy in polynomial time.

$$\mathcal{P}_{SSB} \begin{cases} \min q_{N_r} \\ p_{N_r} = 1 \\ p_D = \sum_{N \in \mathcal{S}(D)} p_N, \forall D \in \mathcal{N}_D \\ p_{(C,N)} p_C = p_N,\ \forall C \in \mathcal{N}_C,\ \forall N \in \mathcal{S}(C) \\ q_T = \sum_{T' \in \mathcal{N}_T} p_{T'} \varphi(o(T), o(T')), \forall T \in \mathcal{N}_T \\ q_C = \sum_{\forall N \in \mathcal{S}(C)} p_{(C,N)} q_N, \forall C \in \mathcal{N}_C \\ q_D \geq q_N, \forall N \in \mathcal{S}(D),\ \forall D \in \mathcal{N}_D \\ p_N \geq 0, \forall N \in \mathcal{N} \\ q_N \in \mathbb{R}, \forall N \in \mathcal{N} \end{cases}$$

We now prove that the determination of an SSB optimal *deterministic* strategy in a decision tree is an NP-hard problem, where the size of the instance is the number of involved decision nodes.

**Theorem 1.** *Finding an optimal deterministic SSB strategy in a decision tree (solving $\mathcal{D}$) is an* NP*-hard problem.*

*Proof.* To ease the presentation, the proof is divided into two parts. While the first part contains all the important ideas of the proof, it proves the result with an SSB utility function which expresses intransitive preferences over the (certain) outcomes of the trees. As this property can seem unnatural or restrictive, the second part of the proof extends the result to an SSB utility function with transitive preferences over the outcomes of the trees.

**Part 1:** The proof relies on a polynomial reduction from 3-SAT, which can be stated as follows:

INSTANCE: a set $X$ of boolean variables, a collection $\mathcal{C}$ of clauses on $X$ such that $|c| = 3$ for every clause $c \in \mathcal{C}$. QUESTION: does there exist an assignment of truth values to the boolean variables of $X$ that satisfies simultaneously all the clauses in $\mathcal{C}$?

Let $X = \{x_1, \ldots, x_n\}$ and $\mathcal{C} = \{c_1, \ldots, c_m\}$. The polynomial generation of a decision tree from an instance of 3-SAT is performed as follows. An example is provided in Figure 2. One defines a decision node $X_i$ for every variable $x_i \in X$. Each node $X_i$ has two children: the first one (chance node denoted by $T_i$) corresponds to the statement "$x_i$ is true" while the second one (chance node denoted by $F_i$) corresponds to "$x_i$ is false". The subset of clauses which includes the positive (resp. negative) literal $x_i$ (resp. $\overline{x_i}$) is denoted by $\{c_{i_1}, \ldots, c_{i_j}\} \subset \mathcal{C}$ (resp. $\{c_{i'_1}, \ldots, c_{i'_k}\} \subset \mathcal{C}$ ). For every clause $c_{i_h}$ (resp. $c_{i'_h}$) one generates a child of $T_i$ (resp. $F_i$) denoted by $c_{i_h}$ (resp. $c_{i'_h}$). These children are terminal nodes. Moreover, one generates an additional child of $T_i$ (resp. $F_i$) denoted by $c_0$, corresponding to a fictive clause. Node $T_i$ (resp. $F_i$) has therefore $j+1$ (resp. $k+1$) children. One adds a chance node $C$ predecessor of all decision nodes $x_i$ and a decision node $D$ as root with $C$ as child. Lastly, for every clause $c_i$ in $\mathcal{C}$ one adds a child $\overline{c}_i$ to the root, a terminal node. The obtained decision tree involves $n+1$ decision nodes, $2n+1$ chance nodes and at most $2n(m+1)+m$ terminal nodes. There is a bijection between the assignments of truth values and the subset of deterministic strategies that choose chance node $C$ at the root: one sets $x_i = 1$ in problem 3-SAT iff edge $(x_i, T_i)$ is included in the strategy, and $x_i = 0$ otherwise. An assignment such that the entire expression is true in 3-SAT corresponds to a strategy such that every clause $c_i$, $i = 1 \ldots m$ is a possible outcome. To complete the reduction, we need to specify the SSB function $\varphi$ and the probabilities in the tree such that property $(\mathcal{P})$ holds:

$(\mathcal{P})$ If the 3-SAT formula is (resp. is not) satisfiable, then any SSB optimal deterministic strategy reaches (resp. does not reach) every clause $c_i$, for $i \in \{1, \ldots, m\}$, with non null probability.

We set the probabilities in the following way. The edges starting from $C$ have probabilities $1/n$. The edges leading to leaves $c_{i=1\ldots m}$ have probabilities $1/m$. Thus, if $c_i$ ($i \in \{1, \ldots, m\}$) is a possible outcome of a strategy then the probability of obtaining $c_i$ is greater than $1/nm$. The SSB function $\varphi$ is defined to have the three following properties. Let $p_{\delta}$ be the lottery resulting from strategy $\delta$. i) If $c_i$ ($i \in \{1, \ldots, m\}$) is reached with a non null probability with strategy $\delta$ then $\varphi(p_{\delta}, \overline{c}_i) > 0$, ii) otherwise $\varphi(p_{\delta}, \overline{c}_i) < 0$. iii) Given two strategies $\delta$ and $\delta'$ both containing edge $(D, C)$, $\varphi(p_{\delta}, p_{\delta'}) = 0$. Indeed, it is easy to see that i), ii) and iii) imply $(\mathcal{P})$. Those conditions are satisfied by the SSB function $\varphi$ defined by: $\forall i, j \in \{0, \ldots, m\}$, $\varphi(c_i, c_j) = 0$;

$\forall i \in \{1,\dots,m\}, \forall j \neq i \in \{0,\dots,m\}, \varphi(\bar{c}_i, c_j) = 1$;
$\forall i \in \{1,\dots,m\}, \varphi(c_i, \bar{c}_i) = nm$.

Indeed, in this case, properties ii) and iii) obviously hold. To see that i) is also satisfied, note that as the probability of obtaining $c_i$ is greater than $1/nm$ if $c_i$ is reached with a non null probability with strategy $\delta$ and $\varphi(c_j, \bar{c}_i) = -1, \forall j \neq i$ by skew symmetry, then:
$\varphi(p_\delta, \bar{c}_i) \geq \varphi(c_i, \bar{c}_i)/nm - (1 - 1/nm) = 1/nm > 0$
where $1 - 1/nm$ is an upper bound on the probability to reach another consequence than $c_i$ with $\delta$.

**Part 2:** Part 1 of the proof uses an SSB utility function with intransitive preferences on the outcomes of the tree (as $\bar{c}_i \succ c_j \succ \bar{c}_j \succ c_i \succ \bar{c}_i$) which can seem unnatural. Thus, we now consider another SSB utility function defined on $\mathbb{R}^2$ by $\varphi(x,y)$ equal to 1 (resp. 0, $-1$) if $x > y$ (resp. $x = y$, $x < y$). The decision tree considered is the same as in the first part of the proof, but now, terminal nodes are replaced by the following lotteries:

$$\begin{aligned} \forall i &\in \{0,\dots,m\} : c_i = (i, 0.5; -i, 0.5) \\ \forall i &\in \{1,\dots,m\} : \\ \bar{c}_i &= (i-0.5, p; -i-0.5, p; m+1, 1-2p) \end{aligned}$$

with $p \in (nm/(1+2nm), 0.5)$. The number of terminal nodes is now upper bounded by $6n(m+1) + 3m$. One can check that with those changes, properties i), ii) and iii) (and therefore property $(\mathcal{P})$) of part 1 still hold. But now, the preferences on the clauses are transitive. □

Note that the associated decision problem "Given $\alpha$, does there exist a deterministic strategy $\delta$ such that $\max_{\delta' \in \Delta} \varphi(p_{\delta'}, p_\delta) \leq \alpha$?" is actually NP-complete. It belongs to NP because, if an oracle provides a strategy $\delta$, one can obtain in polynomial time the value $\max_{\delta' \in \Delta} \varphi(p_{\delta'}, p_\delta)$ by rolling back the decision tree with utility function $V_\delta$ as defined at the beginning of the section. It can be proved NP-hard by using similar arguments as in the proof of Theorem 1.

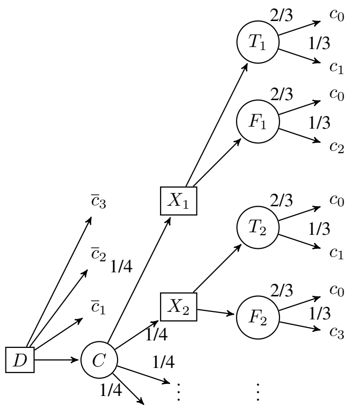

Figure 2: Portion of the decision tree obtained for 3-SAT formula: $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_3 \vee x_4) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_4})$.

## 6 OPTIMIZING THE WEU CRITERION

We first show that, with the WEU criterion, solving optimization problem $(\mathcal{R})$ is equivalent to solving optimization problem $(\mathcal{D})$. We recall that SSB $\equiv$ WEU for $\varphi(p,q) = u(p)w(q) - u(q)w(p)$ and that $\varphi(p_{\delta^*}, p_\delta) \geq 0$, $\forall \delta \in \Delta$ for strategy $\delta^* \in \Delta$ maximizing ratio $u(p_\delta)/w(p_\delta)$ (see Section 3.2).

Let $\Delta = \{\delta_1, \dots, \delta_n\}$. For any strategy $\delta \in \widehat{\Delta}$, there exists positive $\lambda_i$'s summing up to 1 such that $p_\delta = \sum_{i=1}^n \lambda_i p_{\delta_i}$. By bilinearity of $\varphi$, we have:

$$\varphi(p_{\delta_j}, \sum_i \lambda_i p_{\delta_i}) = \sum_{i=1}^n \lambda_i \varphi(p_{\delta_j}, p_{\delta_i}) \quad \forall j \in \{1,\dots,n\} \quad (2)$$

For $\delta_j = \delta^*$, we have $\varphi(p_{\delta_j}, p_{\delta_i}) \geq 0$ for all $i$. Hence, by positivity of $\lambda_i$'s, we can deduce that $\varphi(p_{\delta^*}, \sum_i \lambda_i p_{\delta_i}) \geq 0$ and therefore $p_{\delta^*}$ is preferred to any randomized strategy.

Conversely, consider an optimal strategy $\delta \in \widehat{\Delta}$ for problem $\mathcal{R}$. Let $p_\delta = \sum_i \lambda_i p_{\delta_i}$. In Equation 2, by optimality of $\delta$, we should have $\sum_{i=1}^n \lambda_i \varphi(p_{\delta_j}, p_{\delta_i}) \leq 0$ for all $j \in \{1,\dots,n\}$. For $\delta_j = \delta^*$, it holds that $\varphi(p_{\delta_j}, p_{\delta_i}) \geq 0$ for all $i$, and therefore $\lambda_i \varphi(p_{\delta_j}, p_{\delta_i}) = 0$ for all $i$. Consequently, $\lambda_i > 0 \Rightarrow \varphi(p_{\delta_j}, p_{\delta_i}) = 0$. Stated differently, it means that all deterministic strategies $\delta_i$ such that $\lambda_i > 0$ are optimal for problem $(\mathcal{D})$.

Therefore, for the WEU criterion, solving $(\mathcal{D})$ could be performed with the three following steps: 1) solve $(\mathcal{R})$ with program $\mathcal{P}_{SSB}$ to find an optimal randomized strategy $\delta_r^*$; 2) derive from $\delta_r^*$ an equivalent mixed strategy $\delta_m^*$ (i.e., a probability distribution over deterministic strategies which yields the same probability distribution over the outcomes of the tree); 3) pick any deterministic strategy sampled by $\delta_m^*$ with non null probability. This approach could be performed in polynomial time by using a linear programming approach for step 2) (see e.g. Mastin et al., 2015). Instead of developing this approach, we show how to solve directly problem $(\mathcal{D})$ for a WEU criterion as it will reveal simpler and more efficient.

As shown by Equation 1, optimizing a WEU function entails the maximization of ratio $u(p)/w(p)$. Interestingly, fractional programming is a subfield of operational research dedicated to this type of objective functions (Schaible and Ibaraki, 1983; Stancu-Minasian, 2012). Several solution methods have been developed in this domain to optimize objective functions taking the form of a ratio of two linear functions (we recall that in WEU, functions $u$ and $w$ are linear w.r.t. mixtures). We adapt and present below two of these methods: Megiddo's method (1979) and Dinkelbach's method (1967). Megiddo's method yields a procedure that is polynomial in the number of decision nodes of the decision tree. This proves that solving a decision tree w.r.t.

to the WEU model is a polynomial time problem. Both algorithms are based on a simple idea: solving a decision tree w.r.t. the WEU model can be done by using multiple launches of a sub-routine which solves the decision tree according to the EU model using a utility function $V_\lambda$ of the form $V_\lambda(x) = u(x) - \lambda w(x)$. The validity of the algorithms stems from the following observation:

**Observation 1.** *Given $\lambda \in \mathbb{R}$, let $\delta_\lambda$ denote an optimal EU strategy for utility function $V_\lambda$ and $V_\lambda(p_{\delta_\lambda})$ denote its expected utility. Then, the sign of $V_\lambda(p_{\delta_\lambda})$ is determined by the position of $\lambda$ w.r.t. $\lambda^*$ (we recall that $\lambda^*$ is the optimal value of a deterministic strategy w.r.t. WEU): 1) If $\lambda = \lambda^*$, then $V_\lambda(p_{\delta_\lambda}) = 0$ and $\delta_\lambda$ is an optimal deterministic strategy according to WEU. 2) If $\lambda > \lambda^*$, then $V_\lambda(p_{\delta_\lambda}) < 0$. Indeed, there is no strategy $\delta \in \Delta$ such that $u(p_\delta)/w(p_\delta) > \lambda$. 3) If $\lambda < \lambda^*$, then $V_\lambda(p_{\delta_\lambda}) > 0$.*

Determining a WEU optimal strategy thus amounts to the following optimization problem, where $\lambda^*$ is unknown:

$$\max_{\delta \in \Delta} V_{\lambda^*}(p_\delta) \tag{3}$$

### 6.1 Megiddo's Method

The first algorithm is an adaptation of the method proposed by Megiddo (1979) to solve combinatorial optimization problems with rational objective functions. In our setting, this method can be explained as follows.

As the WEU optimal value $\lambda^*$ is unknown, the idea is to solve the decision tree with a partially specified utility function $V_{\lambda^*}$. The method proceeds by interleaving a bi-objective rolling back procedure with an incremental refinement of the specification of $\lambda^*$. A (sub)strategy $\delta$ is evaluated by the two objective functions $u(p_\delta)$ and $w(p_\delta)$. If $\lambda^*$ is known to belong to interval $[\lambda_l, \lambda_u]$, then $V_{\lambda^*}(p_\delta) \in [u(p_\delta) - \lambda_u w(p_\delta), u(p_\delta) - \lambda_l w(p_\delta)]$. When comparing two lotteries $p_\delta$ and $p_{\delta'}$ in the course of the rolling back procedure, several cases can occur:

i) If $u(p_\delta) - \lambda w(p_\delta) \leq$ (resp. $\geq$) $u(p_{\delta'}) - \lambda w(p_{\delta'})$ for $\lambda \in \{\lambda_l, \lambda_u\}$ then $V_\lambda(p_\delta) \leq$ (resp. $\geq$) $V_\lambda(p_{\delta'})$ for $\lambda \in [\lambda_l, \lambda_u]$, therefore $V_{\lambda^*}(p_\delta) \leq (\geq) V_{\lambda^*}(p_{\delta'})$ and strategy $\delta$ (resp. $\delta'$) can be discarded.

ii) Otherwise, there exists a single value $\lambda_{\delta,\delta'} \in [\lambda_l, \lambda_u]$ such that $u(p_\delta) - \lambda_{\delta,\delta'} w(p_\delta) = u(p_{\delta'}) - \lambda_{\delta,\delta'} w(p_{\delta'})$. Let $\delta''$ denote an optimal strategy for utility function $V_{\lambda_{\delta,\delta'}}$ (obtained by rolling back). Two subcases can occur:

– If $V_{\lambda_{\delta,\delta'}}(p_{\delta''}) <$ (resp. $>$) 0, then $\lambda >$ (resp. $<$) $\lambda^*$ by Observation 1 and interval $[\lambda_l, \lambda_u]$ is updated to $[\lambda_l, \lambda_{\delta,\delta'}]$ (resp. $[\lambda_{\delta,\delta'}, \lambda_u]$). After this update, either $u(p_\delta) - \lambda w(p_\delta) \leq u(p_{\delta'}) - \lambda w(p_{\delta'})$ for $\lambda \in \{\lambda_l, \lambda_u\}$ or $u(p_\delta) - \lambda w(p_\delta) \geq u(p_{\delta'}) - \lambda w(p_{\delta'})$ for $\lambda \in \{\lambda_l, \lambda_u\}$ which brings us back to case i).

– If $V_{\lambda_{\delta,\delta'}}(p_{\delta''}) = 0$, then $\lambda_{\delta,\delta'} = \lambda^*$ and $\delta''$ is an optimal WEU strategy. Thus, the method can be stopped.

As the number of comparisons of lotteries in the rolling back method is upper bounded by $|\mathcal{N}_C| + |\mathcal{N}_T|$, the number of standard rolling back methods (in $O(|\mathcal{N}|)$) launched by Megiddo's method to find an optimal WEU strategy is also upper bounded by $|\mathcal{N}_C| + |\mathcal{N}_T|$. The complexity of the method is therefore $O(|\mathcal{N}^2|)$.

In practice, the initial values of variables $\lambda_l$ and $\lambda_u$ may impact the performance of the method. For instance, a "good" initialization can be obtained by computing the optimal deterministic strategy $\delta_u$ (resp. $\delta_w$) maximizing (resp. minimizing) EU with utility function $u$ (resp. $w$). Variable $\lambda_l$ can then be set to $u(\delta_u)/w(\delta_u)$ and $\lambda_u$ to $\max\{0, u(\delta_u)/w(\delta_w)\}$.

### 6.2 Dinkelbach's method

The second algorithm has not the polynomial time guarantee of the first one, but is more efficient in practice. Let $RB(V)$ denote an optimal EU strategy in $\Delta$ obtained by rolling back the decision tree with utility function $V$ on the leaves (RB for "Rolling Back"). The optimization problem (3) can be solved by computing a sequence of strategies in $\Delta$ through recursive equation:

$$\delta_{t+1} = RB(V_{v(p_{\delta_t})})$$

where $v(p_{\delta_t}) = u(p_{\delta_t})/w(p_{\delta_t})$. A direct corollary from Observation 1 is that while $v(p_{\delta_t}) < \lambda^*$, $v(p_{\delta_{t+1}}) > v(p_{\delta_t})$. Note that, by definition of $\lambda^*$, we cannot have $v(p_{\delta_t}) > \lambda^*$. Therefore, the sequence $(v(p_{\delta_t}))_{t \in \mathbb{N}}$ is strictly increasing until reaching $\lambda^*$. Value $\lambda^*$ is always reached after a finite number of iterations as there is a finite number of values in $\{v(p_\delta) : \delta \in \Delta\}$. After a finite number of iterations, we will thus have $v(p_{\delta_t}) = v(p_{\delta_{t+1}})$ which means that an optimal WEU strategy has been found. While the initial strategy $\delta_0$ can be any feasible strategy in $\Delta$, a "good" choice of $\delta_0$ may increase the efficiency of the approach. For instance, $\delta_0$ can be chosen as an optimal EU strategy according to utility function $V_0$.

## 7 NUMERICAL TESTS

We now present the results of our numerical test[2].

**Random instances.** Our tests were performed on complete binary decision trees. The depth of these decision trees varies from 5 to 20 (21 to 349525 decision nodes), with an alternation of decision nodes and chance nodes. At every terminal node $T$, outcome $o(T)$ is uniformly drawn in interval $[0, 500]$. The mapping $u$ (resp. $w$) from $[0, 500]$ to $[0, 100]$ (resp. $[1, 100]$) is randomly gen-

[2]Methods coded in C++ using Gurobi 5.6.3 for LPs. Times are wall-clocked on a 2.4 GHz Intel Core i5 with 8GB RAM.

| | depth | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|
| DIN | time (sec) | <0.001 | <0.001 | 0.019 | 0.91 |
| | nbsr | 2.2 | 2.98 | 3.1 | 3.66 |
| DIN$^{WI}$ | time (sec) | <0.001 | <0.001 | 0.019 | 0.789 |
| | nbsr | 3.04 | 3.72 | 4.06 | 4.24 |
| MEG | time (sec) | <0.001 | <0.001 | 0.039 | 3.572 |
| | nbsr | 2.02 | 2.74 | 5.52 | 14.38 |
| MEG$^{WI}$ | time (sec) | <0.001 | 0.001 | 0.074 | 4.77 |
| | nbsr | 0.52 | 6.02 | 11.02 | 19.16 |

Table 3: Computation times and number of sub-routine (nbsr in the table) launched for our solution procedures for WEU.

erated while enforcing nondecreasingness (resp. nonincreasingness). Imposing these monotonicity conditions on $u$ and $w$ ensures that $v = u/w$ is a nondecreasing function of outcomes. Table 3 presents the performances of Dinkelbach's method and Megiddo's method (denoted by DIN and MEG from now on) as the depth of the decision tree increases. For each depth level, we give the average computation time over 50 instances in seconds as well as the average number of times the sub-routine rolling back the decision tree for a given utility function $V_\lambda$ was used. Initialization of both methods where performed as described in the previous section.

The results show that both methods are very efficient, solving trees of depth 20 in less than 1 sec (resp. 4 sec) for DIN (resp. MEG ). Method DIN seems to perform best as it requires to launch less sub-routine algorithms.

To measure the impact of the quality of the initialization, we computed the same results for these two methods launched with weak initializations. They are denoted by DIN$^{WI}$ and MEG$^{WI}$ in the table ($WI$ for weak initialization). While DIN$^{WI}$ is initialized with a random initial strategy $\delta_0$, MEG$^{WI}$ starts with the loose lower and upper bounds $\lambda_l = 0$ and $\lambda_u = u(500)/w(0)$.

We observe that a weak initialization does not seem to highly impact the performance of the two methods (DIN$^{WI}$ even achieves better performances than DIN on decision trees of depth 20).

We also tested solving $\mathcal{P}_{SSB}$ on the same trees with SSB utility function $\varphi$ defined by $\varphi(x, y) = 1$ (resp. 0,-1) if $x > y$ (resp. $x = y$, $x < y$). Note that (as illustrated by Example 3), this function may model intransitive preferences. The computation times were averaged over 50 instances. They exponentially increase with the depth of the tree, raising from 2.796 sec for depth 10 to 73.722 sec for depth 12. This is not surprising as $|\mathcal{N}|$ (and thus the number of variables in $\mathcal{P}_{SSB}$) exponentially increases with the depth of the tree.

**Application to Who wants to be a millionaire?** *Who wants to be a millionaire?* is a popular game show, were a contestant must answer a sequence of 15 questions with 4 possible answers. The questions enable the contestant to earn increasing sums of money but are also of increasing difficulties. If the answer given is wrong then the contestant leaves the game with no money except what was earned at the last guarantee point (questions 5 and 10). At each question, the contestant can also decide to stop. She then leaves with the money won so far. We used the Spanish version of the game modeled by Perea and Puerto (2007) where the monetary values of the questions range from 150€ (question 1) to 300000€ (question 15). Lastly, the contestant has three lifelines that can be used once during the game: Phone a friend, 50:50, and Ask the audience. We tested our solution methods for two models of this game. While model 1 (which contains around 70000 nodes) is the original one proposed by Perea and Puerto, model 2 (which contains around 80 million nodes) is a refinement proposed by Jeantet and Spanjaard (2008) to take into account the fact that the candidate may or may not (with a certain probability) know the answer to a given question. The results are averaged over 20 instances where functions of the form $u = x^\alpha$ and $w = 300001^\beta - x^\beta$ were used. For each instance, parameters $\alpha$ and $\beta$ were uniformly sampled in (0,1). Our numerical results are given in Table 4. We observe that both methods DIN and MEG are efficient and that they require very few calls to their sub-routine.

| | | model 1 | model 2 |
|---|---|---|---|
| DIN | time (sec) | 0.017 | 73.307 |
| | nbsr | 2.5 | 2.4 |
| MEG | time (sec) | 0.036 | 104.1 |
| | nbsr | 4.1 | 2.6 |

Table 4: Computation times and number of sub-routine (nbsr in the table) launched DIN and MEG .

## 8 CONCLUSION

We showed that determining an SSB optimal strategy in a decision tree is an NP-hard problem while it becomes polynomial when considering randomized strategies. Regarding the special case of WEU, both the deterministic and the randomized settings collapse as there always exists a WEU optimal strategy which is deterministic. Determining such an optimal strategy is polynomial time by instantiating fractional programming methods. As far as we know, it is the first polynomial time complexity result in sequential decision making under risk for a decision model encompassing Allais' paradox.

For future work, it would be interesting to investigate how these results extend to other frameworks (e.g., finite horizon MDPs). It seems that at least pseudo-polynomial solution methods could be obtained by simple adaptations of the methods proposed here to augmented MDPs (where the states are augmented with the accumulated rewards (Liu and Koenig, 2006)).

## References

S. Chew (1983). A generalization of the quasilinear mean with applications to the measurement of income inequality and decision theory resolving the Allais paradox. *Econometrica* :1065–1092.

W. Dinkelbach (1967). On nonlinear fractional programming. *Management science* **13**(7):492–498.

H. Fargier, G. Jeantet, and O. Spanjaard (2011). Resolute choice in sequential decision problems with multiple priors. In *Proceedings of IJCAI 2011*. 2120–2125.

F. Fischer, O. Hudry, and R. Niedermeier (2016). Weighted tournament solutions. In *Handbook of Computational Social Choice*, Cambridge University Press. 85–102.

P. Fishburn (1984a). Dominance in SSB utility theory. *Journal of Economic Theory* **34**(1):130–148.

P. Fishburn (1984b). SSB utility theory: an economic perspective. *Math. Social Sciences* **8**(1):63 – 94.

P. C. Fishburn (1982). Nontransitive measurable utility. *Journal of Mathematical Psychology* **26**:31–67.

P. C. Fishburn (1983). Transitive measurable utility. *Journal of Economic Theory* **31**(2):293–317.

M. Gardner (1970). Mathematical games: The paradox of nontransitive dice and the elusive principle of indifference. *Sci. Amer.* **223**:110–114.

H. Gilbert, O. Spanjaard, P. Viappiani, and P. Weng (2015). Solving MDPs with skew symmetric bilinear utility functions. In *Proc. of IJCAI 2015*. 1989–1995.

H. Gilbert, B. Zanuttini, P. Viappiani, P. Weng, and E. Nicart (2016). Model-free reinforcement learning with skew-symmetric bilinear utilities. In *UAI 2016*.

P. Hammond (1988). Consequentialist foundations for expected utility. *Theory and decision* **25**(1):25–78.

N. Huntley and M. C. Troffaes (2008). An efficient normal form solution to decision trees with lower previsions. In *Soft Methods for Handling Variability and Imprecision*, Springer. 419–426.

J.-Y. Jaffray (1998). Implementing resolute choice under uncertainty. In *Proceedings of UAI 1998*. Morgan Kaufmann Publishers Inc., 282–288.

J.-Y. Jaffray and M. Jeleva (2007). Information processing under imprecise risk with the Hurwicz criterion. In *Proc. of ISIPTA 2007*. 233–242.

G. Jeantet, P. Perny, and O. Spanjaard (2012). Sequential decision making with rank dependent utility: a minimax regret approach. In *AAAI 2012*. 1931–1937.

G. Jeantet and O. Spanjaard (2008). Rank-dependent probability weighting in sequential decision problems under uncertainty. In *Proc. of ICAPS 2008*. 148–155.

G. Jeantet and O. Spanjaard (2009). Optimizing the Hurwicz criterion in decision trees with imprecise probabilities. In *Proc. of ADT 2009*. 340–352.

D. Kahneman and A. Tversky (1979). Prospect theory: An analysis of decisions under risk. *Econometrica* :263–291.

L. G. Khachiyan (1980). Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics* **20**(1):53–72.

D. Kikuti, F. G. Cozman, and R. Shirota Filho (2011). Sequential decision making with partially ordered preferences. *Artif. Intelligence* **175**(7-8):1346–1365.

Y. Liu and S. Koenig (2006). Functional value iteration for decision-theoretic planning with general utility functions. In *Proc. of AAAI 2006*. vol. 21, 1186.

A. Mastin, P. Jaillet, and S. Chin (2015). Randomized minmax regret for combinatorial optimization under uncertainty. In *Proc. of ISAAC 2015*. 491–501.

E. F. McClennen (1990). *Rationality and dynamic choice: Foundational explorations*. Cambridge university press.

N. Megiddo (1979). Combinatorial optimization with rational objective functions. *Math. Oper. Res.* **4**(4):414–424.

J. von Neumann and O. Morgenstern (1947). *Theory of games and economic behaviour*. Princeton Univ Press.

E. Nicart, B. Zanuttini, H. Gilbert, B. Grilhères, and F. Praca (2016). Building document treatment chains using reinforcement learning and intuitive feedback. In *Proc. of ICTAI 2016*. 635–639.

T. D. Nielsen and J.-Y. Jaffray (2006). Dynamic decision making without expected utility: An operational approach. *European J. of Op. Research* **169**(1):226–246.

F. Perea and J. Puerto (2007). Dynamic programming analysis of the TV game "Who wants to be a millionaire?". *European J. of Op. Research* **183**(2):805–811.

J. Quiggin (1993). *Generalized expected utility theory: the rank-dependent model*. Kluwer.

S. Schaible and T. Ibaraki (1983). Fractional programming. *European J. of Op. Research* **12**(4):325–338.

I. Stancu-Minasian (2012). *Fractional programming: theory, methods and applications*, vol. 409. Springer Science & Business Media.

A. Tversky (1969). Intransitivity of preferences. *Preference, Belief, and Similarity* :433.

H. Young (1977). Extending Condorcet's rule. *Journal of Economic Theory* **16**(2):335–353.