Proceedings of Machine Learning Research 147:235–243, 2021 ISIPTA 2021

Improving Algorithms for Decision Making with the Hurwicz Criterion

Nawapon Nakharutai NAWAPON.NAKHARUTAI@CMU.AC.TH
Department of Statistics, Faculty of Science, Chiang Mai University, Thailand

Matthias C. M. Troffaes MATTHIAS.TROFFAES@DURHAM.AC.UK
Department of Mathematical Sciences, Durham University, United Kingdom

Abstract

We propose two improved algorithms for evaluating the Hurwicz criterion in the context of decision making with lower previsions, along with a new benchmarking algorithm for measuring these improvements.

The Hurwicz criterion is a well-known criterion for decision making with lower previsions under severe uncertainty when decision makers want to balance between pessimistic and optimistic extremes. When the domain of the lower prevision, the set of possible outcomes and the set of possible decisions are all finite, the classic method for applying this criterion goes by solving a sequence of linear programs. We show how to improve this classic algorithm, based on similar improvements that we have proposed for other decision criteria. Additionally, to allow benchmarking these improvements, we provide a new algorithm for randomly generating artificial decision problems with a set number of Hurwicz gambles.

In our simulation, our proposed algorithms for Hurwicz outperform the standard algorithm in most scenarios except when the set of outcomes is small, the domain of the lower prevision is large, and there are many Hurwicz optimal decisions at once, in which case our proposed algorithms are slightly slower.

Keywords: lower prevision, decision, Hurwicz, algorithm, primal-dual method, linear program

1. Introduction

Consider a decision problem where one wants to choose the best option from a set of all possible options. Selecting an option will yield an uncertain reward which also depends on uncertain states of nature. Rewards are assumed to be expressed in utility scale. The uncertain reward can be viewed as a bounded real-valued function, called *gamble*, on a set of possible outcomes. Therefore, a decision maker will select gambles from a set of possible gambles.

In a classical decision theory, if precise probabilities for all events are known, then it is reasonable that the decision maker will choose a gamble that leads to the highest expected utility [1]. However, when these precise probabilities are not completely known, one way to handle this issue is to use *lower previsions* which correspond to expectation bounds [20]. In this paper, we consider decision problems using lower previsions.

Several decision criteria with lower previsions have been studied and developed by many authors [7, 20, 16, 17, 4]. The Γ-maximin criterion selects the decision which maximises the lower prevision (i.e. the worst case expected utility), while the Γ-maximax criterion selects the decision which maximises the upper prevision (i.e. the best case expected utility) [17]. When the decision maker wants to balance between the pessimistic and the optimistic cases, the decision maker can apply the *Hurwicz* criterion (also known as Γ-maximix) which selects the decision that maximises a convex combination of the lower prevision and the upper prevision [4]. Note that these three criteria will usually return a single gamble. There are other decision criteria that return a set of gambles, for example, *maximality* and *interval dominance* which are based on strict partial preference orders [17].

Several authors, including Gilboa and Schmeidler [3], Walley [20], Zaffalon et al. [23], Seidenfeld [16] and Troffaes [17], have studied decision criteria for lower previsions. In addition, many authors have also studied algorithms for them. For example, Kikuti et al. [6], Matt [8], Jansen et al. [5] Nakharutai et al. [14] and Troffaes and Hable [19, p. 336] have studied and improved algorithms for maximality and interval dominance. Kikuti et al. [6] Troffaes and Hable [19, p. 335] and Nakharutai et al. [15] have studied algorithms for Γ-maximin and Γ-maximax. From these, [14] and [15] performed a comparative benchmarking study of their algorithms. However, as far as we known, in the context of lower previsions, there is no in-depth study of algorithms for Hurwicz in the literature.

Determining whether a gamble is optimal under Γ-maximin, Γ-maximax, or Hurwicz, can be done by calculating the value of the lower and/or upper previsions for all gambles in the set. If the number of gambles in the set, the number of outcomes and the domain of lower prevision are finite, then one can solve linear programs to compute the value of the lower previsions [6, 19, 5]. Note that there could be scenarios where non-linear programming can be applied, for instance, under certain structural assumptions such as independence, however, this is beyond the scope of this paper.

In [12, 13] we investigated how to efficiently solve linear programs for evaluating lower previsions. Based on these results, we further studied how to improve and benchmark algorithms for maximality and interval dominance in [14] and for Γ-maximin and Γ-maximax and interval dominance in [15]. In this paper, we investigate how several proposed improvements can be applied to the Hurwicz criterion.

The contributions of this work are as follows. We propose two new improved algorithms for the Hurwicz criterion. In these algorithms, we apply a technique to very quickly get initial feasible starting points for all linear programs that we need to solve (based on earlier work [12, 13]), and we propose an early stopping criterion for detecting non-Hurwicz gambles early on the algorithm, to save computational effort. To benchmark these new algorithms, we propose a new algorithm for randomly generating artificial sets of gambles with a pre-specified number of Hurwicz gambles.

The paper is organised as follows. In Section 2, we briefly review lower previsions, natural extension, and the decision criteria that will be used throughout the rest of the paper. In Section 3, we propose two new algorithms for the Hurwicz criterion. In Section 4, we provide an algorithm to randomly generate a set of gambles with a precise number of Hurwicz gambles, which can be used for benchmarking algorithms for Hurwicz. We also compare the performance of different algorithms for Hurwicz on these generated sets. Finally, we conclude the paper in Section 5.

2. Preliminaries

In this section, we review lower previsions, natural extension and two decision criteria: Hurwicz and interval dominance, which we will need later. For further discussion about these criteria and other decision criteria e.g. Γ-maximin, Γ-maximax, maximality and E-admissibility, we refer to [7, 20, 16, 17, 4] and references therein.

2.1. Lower Previsions and Natural Extension

We denote the set of possible outcomes by Ω. A gamble, typically denoted by f, is a bounded real-valued function $\Omega \to \mathbb{R}$ and represents an uncertain payoff, that is, we receive $f(\omega)$ when $\omega \in \Omega$ is revealed as the true outcome. We also denote the set of all gambles by $\mathscr{L}$. Following [21, 22, 20], a subject, such as a decision maker, can model their uncertainty through a so-called *lower prevision* $\underline{P}$ which is a real-valued function defined on a domain $\operatorname{dom}\underline{P} \subseteq \mathscr{L}$. Specifically, given a gamble $f \in \operatorname{dom}\underline{P}$, we view $\underline{P}(f)$ as the subject's supremum buying price for f. Its conjugate upper prevision $\overline{P}$ on $\operatorname{dom}\overline{P} := \{-f\colon f \in \operatorname{dom}\underline{P}\}$ is defined by $\overline{P}(f) := -\underline{P}(-f)$ and represents the subject's infimum selling price for f [18, p. 41]. It has been extensively argued that lower and upper previsions are suitable for modelling uncertainty especially when little information is available [20, 9, 10, 18].

In this paper, we assume that every lower prevision considered *avoids sure loss*, namely, for all $n \in \mathbb{N}$, all non-negative $\lambda_1, \ldots, \lambda_n$, and all $f_1, \ldots, f_n \in \operatorname{dom}\underline{P}$, we require that [18, p. 42]:

$$\sup_{\omega\in\Omega}\left(\sum_{i=1}^{n}\lambda_i\,[f_i(\omega)-\underline{P}(f_i)]\right) \geq 0. \tag{1}$$

Otherwise, we can find combinations of gambles for which the subject is willing to pay more than what she could ever gain, which makes no sense [18, p. 44].

The *natural extension* $\underline{E}$ of $\underline{P}$ extends $\underline{P}$ to a lower prevision defined on all gambles. For every gamble $f \in \mathscr{L}$, it is defined by [18, p. 47]:

$$\underline{E}(f) := \sup\Big\{\alpha\in\mathbb{R}\colon f-\alpha \geq \sum_{i=1}^{n}\lambda_i(f_i-\underline{P}(f_i)),\; n\in\mathbb{N},\, f_i\in\operatorname{dom}\underline{P},\, \lambda_i\geq 0\Big\}, \tag{2}$$

and one can show that this is also a lower prevision (i.e. that the supremum is finite) provided $\underline{P}$ avoids sure loss as assumed [18, p. 68]. It is the supremum price that a subject should be willing to pay for f, given the supremum buying prices $\underline{P}(f_i)$ for all $f_i \in \operatorname{dom}\underline{P}$. Its conjugate is denoted by $\overline{E}(f) := -\underline{E}(-f)$.

Throughout this study, we only consider the case that both Ω and $\operatorname{dom}\underline{P}$ are finite, so $\underline{E}(f)$ can be obtained by solving a linear program [19, p. 331].

2.2. Decision Criteria

As we mentioned before, there are many criteria for decision making with lower previsions. Here, we will discuss Hurwicz and three other criteria that are related to Hurwicz, namely, Γ-maximin, Γ-maximax and interval dominance.

Let $\mathscr{K} \subseteq \mathscr{L}$ be a set of gambles. We first consider the Γ-maximin criterion:

Definition 1 *[17] The set of* Γ*-maximin gambles of* $\mathscr{K}$ *is defined by*

$$\operatorname{opt}_{\underline{E}}(\mathscr{K}) := \arg\max_{f\in\mathscr{K}} \underline{E}(f). \tag{3}$$

The Γ-maximin criterion selects a gamble that maximises the lower natural extension. In contrast, the following criterion simply selects a gamble that maximises the upper natural extension:

Definition 2 *[17] The set of* Γ*-maximax gambles of* $\mathscr{K}$ *is defined by*

$$\operatorname{opt}_{\overline{E}}(\mathscr{K}) := \arg\max_{f\in\mathscr{K}} \overline{E}(f). \tag{4}$$

Clearly, Γ-maximin reflects a pessimistic decision maker, while Γ-maximax reflects an optimistic one. To balance between these extremes, we can apply another criterion called Hurwicz (also known as Γ-maximix):

Definition 3 *[4] For any $\beta \in [0,1]$ (fixed by the decision maker), the set of Hurwicz gambles of $\mathscr{K}$ is defined by*

$$\mathrm{opt}_{\beta}(\mathscr{K}) := \arg\max_{f\in\mathscr{K}}(\beta\underline{E}(f) + (1-\beta)\overline{E}(f)). \quad (5)$$

The Hurwicz criterion chooses a gamble that maximizes a convex combination of the best and worst possible expected payoff.

Finally, we consider *interval dominance* which is based on a strict partial preference order.

Definition 4 *Consider the following strict partial preference order, defined between any two gambles f and $g \in \mathscr{L}$:*

$$f \sqsupset g \text{ if } \underline{E}(f) > \overline{E}(g) \quad (6)$$

The set of interval dominant gambles of $\mathscr{K}$ is defined by

$$\mathrm{opt}_{\sqsupset}(\mathscr{K}) := \{f \in \mathscr{K} : (\forall g \in \mathscr{K})(g \not\sqsupset f)\} \quad (7)$$
$$= \{f \in \mathscr{K} : \overline{E}(f) \geq \max_{g\in\mathscr{K}} \underline{E}(g)\}. \quad (8)$$

Note that the following relationship holds [4]:

$$\mathrm{opt}_{\underline{E}}(\mathscr{K}) \cup \mathrm{opt}_{\overline{E}}(\mathscr{K}) \cup \mathrm{opt}_{\beta}(\mathscr{K}) \subseteq \mathrm{opt}_{\sqsupset}(\mathscr{K}). \quad (9)$$

3. Algorithms

In this section, we discuss how to improve algorithms for Hurwicz based on similar improvements proposed in [14, 15], where we provided improved algorithms for Γ-maximin, Γ-maximax, maximality and interval dominance.

3.1. Base Algorithm for Hurwicz

Let $\mathscr{K}$ be a set of k gambles. To find $\arg\max_{f\in\mathscr{K}}(\beta\underline{E}(f) + (1-\beta)\overline{E}(f))$, we could simply evaluate k lower natural extensions and k upper natural extensions [19, Sec. 16.3.1]; see algorithm 1.

Algorithm 1: Hurwicz [19]
Data: a set of k gambles $\mathscr{K} = \{f_1, \ldots, f_k\}$
Result: the index of a single Hurwicz gamble

```
M ← −∞
for i = 1 : k do
    if βE̲(f_i) + (1−β)Ē(f_i) > M then
        i* ← i & M ← βE̲(f_i) + (1−β)Ē(f_i)
    end
end
return i* // index of a Hurwicz gamble
```

3.2. Improvements for Hurwicz

In general, one can evaluate the natural extension by solving a linear program. There are several common linear programming methods, for example, the simplex method, the affine scaling method, and the primal-dual interior point method. These methods are iterative algorithms, i.e., they produce a sequence of points that converge to an optimal feasible solution.

As suggested in [13, 14, 15], one of the most effective ways to evaluate $\underline{E}(f)$ goes by the primal-dual interior point method, which simultaneously solves the following pair of linear programs [14]:

$$\text{(P1)} \quad \min \sum_{\omega\in\Omega} f(\omega)p(\omega) \quad (10)$$
$$\text{subject to} \quad \forall f_i \in \mathrm{dom}\,\underline{P}\colon \sum_{\omega\in\Omega}(f_i(\omega) - \underline{P}(f_i))p(\omega) \geq 0 \quad (11)$$
$$\sum_{\omega\in\Omega} p(\omega) = 1 \quad (12)$$
$$\forall\omega\colon p(\omega) \geq 0, \quad (13)$$

$$\text{(D1)} \quad \max \ \alpha \quad (14)$$
$$\text{subject to} \quad \forall\omega\in\Omega\colon \sum_{i=1}^{n}(f_i(\omega) - \underline{P}(f_i))\lambda_i + \alpha \leq f(\omega) \quad (15)$$
$$\forall i\colon \lambda_i \geq 0 \quad (\alpha \text{ free}). \quad (16)$$

To evaluate $\overline{E}(f)$, note that $\overline{E}(f) = -\underline{E}(-f)$ which is equivalent to solving the following pair of linear programs [15]:

$$\text{(P2)} \quad \min \ \beta \quad (17)$$
$$\text{subject to} \quad \forall\omega\in\Omega\colon \beta - \sum_{i=1}^{k}(f_i(\omega) - \underline{P}(f_i))\lambda_i \geq f(\omega) \quad (18)$$
$$\forall i\colon \lambda_i \geq 0 \quad (\beta \text{ free}), \quad (19)$$

$$\text{(D2)} \quad \max \sum_{\omega\in\Omega} f(\omega)p(\omega) \quad (20)$$
$$\text{subject to} \quad \forall f_i \in \mathrm{dom}\,\underline{P}\colon \sum_{\omega\in\Omega}(f_i(\omega) - \underline{P}(f_i))p(\omega) \geq 0 \quad (21)$$
$$\sum_{\omega\in\Omega} p(\omega) = 1 \quad (22)$$
$$\forall\omega\colon p(\omega) \geq 0. \quad (23)$$

In that work, two improvements were proposed, namely, an efficient way to obtain initial feasible starting points and an early stopping criterion. Specifically, for (P1) and (D2), one can easily obtain a common (i.e. independent of f) initial feasible starting point by applying the first phase of the two-phase method [13, §4.2]. To obtain an initial feasible solution for (D1) and (P2), we can apply a result from Nakharutai et al. [13, Theorem 7]. Furthermore, starting with these feasible points, the primal-dual method can generate a sequence of feasible points converging to an optimal solution [2, §7.3]. This is useful because feasibility is required to allow us to apply the early stopping criterion, whilst the standard primal-dual interior point method does not produce a sequence of feasible points, unless we start from a feasible point.

Next, we note that we can exploit the fact that, in algorithm 1, we only need to verify whether or not $\beta\underline{E}(f_i) + (1-\beta)\overline{E}(f_i) > M$, where $M = \max_{1\leq j\leq i-1}\beta\underline{E}(f_j) + (1-\beta)\overline{E}(f_j)$. Let $\underline{\ell}_i$ and $\underline{u}_i$ be the current lower and upper bounds for $\underline{E}(f_i)$, and let $\overline{\ell}_i$ and $\overline{u}_i$ be the current lower and upper bounds for $\overline{E}(f_i)$. Note that

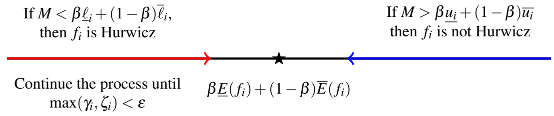

Figure 1: Early stopping criterion for Hurwicz

$$\beta\underline{\ell}_i + (1-\beta)\overline{\ell}_i \leq \beta\underline{E}(f_i) + (1-\beta)\overline{E}(f_i) \leq \beta\underline{u}_i + (1-\beta)\overline{u}_i \quad (24)$$

Clearly, given these bounds, one can stop as soon as $M > \beta\underline{u}_i + (1-\beta)\overline{u}_i$ because in this case, f_i cannot be Hurwicz. On the other hand, if $M < \beta\underline{\ell}_i + (1-\beta)\overline{\ell}_i$, then f_i is Hurwicz among the gambles up to index i. However, in this case, we still have to continue evaluating the linear programs until convergence to obtain the optimal values, as we need to know $\beta\underline{E}(f_i) + (1-\beta)\overline{E}(f_i)$ to assign it to be the new value for M. Figure 1 illustrates this argument.

We also observe that if we can identify a Hurwicz gamble (across all gambles in $\mathscr{K}$) early on, then it is more likely that we can stop early in later stages of the algorithm, because a higher value of M translates into fewer iterations of the algorithm with the early stopping criterion kicking in more often. Following [14, 15], if we sort all gambles in $\mathscr{K}$ as $f_1, \dots, f_k$ such that for some probability mass function p, for all $i < j$:

$$E_p(f_i) \geq E_p(f_j), \quad (25)$$

then it is more likely that for a given β, $\beta\underline{E}(f_i) + (1-\beta)\overline{E}(f_i) \geq \beta\underline{E}(f_j) + (1-\beta)\overline{E}(f_j)$. Therefore, there is a high chance that we will find a gamble that is Hurwicz with a smaller index. Even though this sorting does not always guarantee this, it does not require much extra computation, and therefore it is worth to implement this trick at the initialization. An algorithm that implements all these suggestions is presented in algorithm 2.

In that algorithm, we use $\beta\gamma_i + (1-\beta)\zeta_i$ to bound the error. Indeed, the error is bounded by the difference between the left and right hand sides of eq. (24), that is,

$$\begin{aligned}\beta\underline{u}_i + (1-\beta)\overline{u}_i - \beta\underline{\ell}_i - (1-\beta)\overline{\ell}_i &= \beta(\underline{u}_i - \underline{\ell}_i) + (1-\beta)(\overline{u}_i - \overline{\ell}_i) \\ &= \beta\gamma_i + (1-\beta)\zeta_i \quad (26)\end{aligned}$$

3.3. Further Improving Hurwicz

In [15], we suggested an efficient way to find all Γ-maximin (or Γ-maximax) gambles, whilst also making it possible to remove non Γ-maximin (or non Γ-maximax) gambles before obtaining the value of $\underline{E}(f_1)$ (or $\overline{E}(f_1)$). We can treat the Hurwicz criterion similarly.

Consider the following toy example. Let $\mathscr{K} = \{f_1, f_2, f_3, f_4\}$ be a set of gambles with $\underline{E}(f_i)$ and $\overline{E}(f_i)$ given as in fig. 2. Suppose that we want to find all Hurwicz

Algorithm 2: Hurwicz
Data: a set of k gambles $\mathscr{K} = \{f_1, \dots, f_k\}$ such that $E_p(f_1) \geq E_p(f_2) \geq \dots \geq E_p(f_k)$ for some p that is a feasible solution of (P1); a small number ε; initial feasible states x_i^P and x_i^D for (P1) and (D1) and initial feasible states y_i^P and y_i^D for (P2) and (D2) corresponding to f_i respectively;
Result: a single Hurwicz gamble

```
M ← βE̲(f_1) + (1 − β)Ē(f_1); i* ← 1
for i = 2 : k do
    repeat
        (x_i^P, x_i^D) ← φ(f_i, x_i^P, x_i^D) // next iteration to update E̲(f_i)
        ℓ̲_i ← e̲_*(f_i, x_i^P, x_i^D)   // lower bound for E̲(f_i)
        u̲_i ← e̲^*(f_i, x_i^P, x_i^D) // upper bound for E̲(f_i)
        γ_i = u̲_i − ℓ̲_i
        (y_i^P, y_i^D) ← ψ(f_i, y_i^P, y_i^D) // next iteration to update Ē(f_i)
        ℓ̄_i ← ē_*(f_i, y_i^P, y_i^D) // lower bound for Ē(f_i)
        ū_i ← ē^*(f_i, y_i^P, y_i^D)   // upper bound for Ē(f_i)
        ζ_i = ū_i − ℓ̄_i
    until M > βu̲_i + (1 − β)ū_i or βγ_i + (1 − β)ζ_i ≤ ε;
    if M < βℓ̲_i + (1 − β)ū_i then
        M ← βℓ̲_i + (1 − β)ū_i; i* ← i
    end
end
return i* // index of a Hurwicz gamble
```

gambles (according to fig. 2, f_1 and f_4 are both Hurwicz). To start, given initial states (x_i^P, x_i^D) for each i, we could first calculate $\underline{\ell}_i := \underline{e}_*(f_i, x_i^P, x_i^D)$ and $\underline{u}_i := \underline{e}^*(f_i, x_i^P, x_i^D)$. Similarly, given initial states (y_i^P, y_i^D) for each i, we can also calculate $\overline{\ell}_i := \overline{e}_*(f_i, y_i^P, y_i^D)$ and $\overline{u}_i := \overline{e}^*(f_i, y_i^P, y_i^D)$. Suppose that the result is as in fig. 2. The dashed line represents $\max_{i \in R} \beta\underline{\ell}_i + (1-\beta)\overline{\ell}_i$. For each f_i, we compute $\beta\underline{u}_i + (1-\beta)\overline{u}_i$, where they are represented by ◯ in fig. 2. Note that $\beta\underline{u}_2 + (1-\beta)\overline{u}_2$ and $\beta\underline{u}_3 + (1-\beta)\overline{u}_3$ are smaller than $\max_{i \in R} \beta\underline{\ell}_i + (1-\beta)\overline{\ell}_i$. Therefore, we immediately know that f_2 and f_3 cannot be Hurwicz. At this point, we can quickly eliminate these non Hurwicz gambles.

We now generalize this example to construct an algorithm that can sequentially narrow a set $R \subseteq \{1, \dots, k\}$ of potentially Hurwicz gambles (or rather, their indices). To do so, for each gamble f_i in $\mathscr{K}$, (i) we evaluate bounds for $\underline{E}(f_i)$, namely, $\underline{\ell}_i := \underline{e}_*(f_i, x_i^P, x_i^D)$ and $\underline{u}_i := \underline{e}^*(f_i, x_i^P, x_i^D)$, and (ii) we also evaluate bounds for $\overline{E}(f_i)$, namely, $\overline{\ell}_i := \overline{e}_*(f_i, y_i^P, y_i^D)$ and $\overline{u}_i := \overline{e}^*(f_i, y_i^P, y_i^D)$. Next, we calculate $M_* := \max_{i \in R} \beta\underline{\ell}_i + (1-\beta)\overline{\ell}_i$ and $M^* := \max_{i \in R} \beta\underline{u}_i + (1-\beta)\overline{u}_i$. Note that M_* and M^* bound the Hurwicz value, that is

$$M_* \leq \max_{f \in \mathscr{K}} (\beta\underline{E}(f) + (1-\beta)\overline{E}(f)) \leq M^* \quad (27)$$

Then, by eq. (24), any gamble f_i for which $\beta\underline{u}_i + (1-\beta)\overline{u}_i < M_*$ will not be Hurwicz and therefore can be eliminated. Then, we update the states (x_i^P, x_i^D) and (y_i^P, y_i^D) for

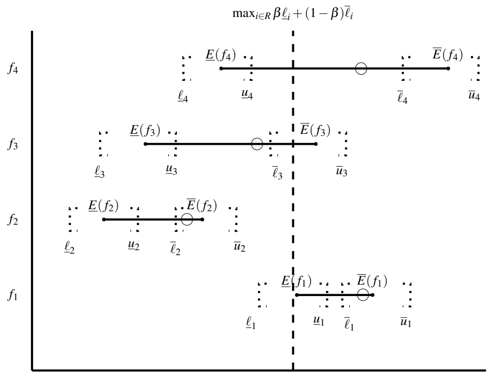

Figure 2: Early eliminating non Hurwicz gambles

Gambles	$\underline{E}(f_i)$	$\overline{E}(f_i)$	$\beta\underline{E}(f_i)+(1-\beta)\overline{E}(f_i)$
f_1	7	9	8
f_2	2	4.5	3.25
f_3	3	8	5.25
f_4	5	11	8

Table 1: Example of different gambles in fig. 2 providing their $\underline{E}(f_i)$ and $\overline{E}(f_i)$ with $\beta = 0.5$.

all gambles f_i that are still potentially Hurwicz. The process will be repeated until either only one gamble is left in the set of potentially Hurwicz gambles, or until all errors are less than some given tolerance ε, in which case we have found multiple Hurwicz gambles. An algorithm that implements these strategies is presented in algorithm 3.

To summarise, using similar arguments as those proposed in [14, 15], we have proposed two new algorithms for finding Hurwicz gambles (algorithms 2 and 3). The main differences between algorithms 2 and 3 occur in the processing and in the output. In particular, an implementation of algorithm 2 must necessarily process gambles sequentially, whilst an implementation of algorithm 3 can exploit parallel processing. Moreover, the output of algorithm 2 gives only a single Hurwicz gamble whilst the output of algorithm 3 gives an index set of all Hurwicz gambles.

Algorithm 3: Hurwicz

Data: a set of k gambles $\mathcal{K}=\{f_1,\dots,f_k\}$; a small number ε; initial feasible states x_i^P and x_i^D for (P1) and (D1) and initial feasible states y_i^P and y_i^D for (P2) and (D2) corresponding to f_i respectively;

Result: a set of Hurwicz gambles

$R \leftarrow \{1,2,\dots,k\}$

repeat

$\quad \forall i\in R: (x_i^P,x_i^D)\leftarrow \phi(f_i,x_i^P,x_i^D)$ // next iteration to update $\underline{E}(f_i)$

$\qquad \underline{\ell}_i \leftarrow \underline{e}_*(f_i,x_i^P,x_i^D)$ // lower bound for $\underline{E}(f_i)$

$\qquad \underline{u}_i \leftarrow \underline{e}^*(f_i,x_i^P,x_i^D)$ // upper bound for $\underline{E}(f_i)$

$\qquad \gamma_i = \underline{u}_i - \underline{\ell}_i$

$\qquad (y_i^P,y_i^D)\leftarrow \psi(f_i,y_i^P,y_i^D)$, // next iteration to update $\overline{E}(f_i)$

$\qquad \overline{\ell}_i \leftarrow \overline{e}_*(f_i,y_i^P,y_i^D)$ // lower bound for $\overline{E}(f_i)$

$\qquad \overline{u}_i \leftarrow \overline{e}^*(f_i,y_i^P,y_i^D)$ // upper bound for $\overline{E}(f_i)$

$\qquad \zeta_i = \overline{u}_i - \overline{\ell}_i$

$\quad M_* \leftarrow \max_{i\in R}\beta\underline{\ell}_i+(1-\beta)\overline{\ell}_i$

$\quad M^* \leftarrow \max_{i\in R}\beta\underline{u}_i+(1-\beta)\overline{u}_i$

$\quad R \leftarrow \{i\in R: \beta\underline{u}_i+(1-\beta)\overline{u}_i \ge M_*\}$

$\quad \eta \leftarrow \max\{M^*-M_*, \max_{i\in R}\{\beta\gamma_i+(1-\beta)\zeta_i\}\}$

until $|R|=1$ **or** $\eta\le\varepsilon$;

return R // index set of all Hurwicz gambles

4. Benchmarking

4.1. Generating Sets of Gambles with Precise Number of Hurwicz Gambles

Suppose that one would like to benchmark the algorithms for finding Hurwicz gambles on random sets of gambles. One easy way to do is to randomly generate a set of gambles such that for each gamble i and ω, $f_i(\omega)$ is sampled uniformly from $[0,1]$. However, in this way, we do not control the exact number of Hurwicz gambles in the set of generated gambles. To compare different algorithms, it is useful to generate sets of gambles with a precise number of Hurwicz gambles, similar to benchmarking techniques proposed in [14, 11].

To generate a set of k gambles $\mathcal{K}$ such that exact b gambles are Hurwicz, that is, $|\mathrm{opt}_\beta(\mathcal{K})| = b \le k$, a naive idea is first to generate b Hurwicz gambles and then to add $k-b$ non Hurwicz gambles. Specifically, we can start with $\mathcal{K}=\{f\}$, where f is immediately Hurwicz. Next, we randomly generate a gamble h such that

$$\mathrm{opt}_\beta(\mathcal{K}\cup\{h\}) = \mathrm{opt}_\beta(\mathcal{K})\cup\{h\}, \tag{28}$$

and then we add h to $\mathcal{K}$. We continue this process until we obtain b Hurwicz gambles in the set $\mathcal{K}$. Next, we randomly generate a gamble h for which

$$\mathrm{opt}_\beta(\mathcal{K}\cup\{h\}) = \mathrm{opt}_\beta(\mathcal{K}), \tag{29}$$

and then we add h to $\mathcal{K}$. We continue this process until we obtain $k-b$ non Hurwicz gambles in the set $\mathcal{K}$.

However, a gamble h that we randomly generate may not easily match the conditions in eq. (28) or in eq. (29). To address this issue, we modify a gamble h by shifting h for some $\alpha\in\mathbb{R}$ in order to make $h-\alpha$ satisfy either eq. (28) or eq. (29). Ranges for α to modify h appropriately can be directly calculated as follows:

Theorem 5 *Let $\mathcal{K}$ be a set of gambles where the gamble f in $\mathcal{K}$ is Hurwicz. Let h be another gamble, let $\alpha\in\mathbb{R}$, and let $\mathcal{K}' := \mathcal{K}\cup\{h-\alpha\}$. Define*

$$\alpha^* := \beta(\underline{E}(h)-\underline{E}(f)) + (1-\beta)(\overline{E}(h)-\overline{E}(f)) \tag{30}$$

Then

(i) $h-\alpha$ *and* f *are both Hurwicz in* $\mathscr{K}'$ *if* $\alpha=\alpha^*$.

(ii) $h-\alpha$ *is Hurwicz in* $\mathscr{K}'$ *but not* f *if* $\alpha<\alpha^*$.

(iii) $h-\alpha$ *is not Hurwicz in* $\mathscr{K}'$ *but* f *is if* $\alpha>\alpha^*$.

Proof Note that $\alpha^*=\max_{g\in\mathscr{K}}(\beta\underline{E}(g)+(1-\beta)\overline{E}(g))$ since f is a Hurwicz gamble in $\mathscr{K}$. From the definition of Hurwicz, $h-\alpha$ is also Hurwicz if and only if $\beta\underline{E}(h-\alpha)+(1-\beta)\overline{E}(h-\alpha)=\max_{g\in\mathscr{K}}(\beta\underline{E}(g)+(1-\beta)\overline{E}(g))$. This is equivalent to $\beta\underline{E}(h-\alpha)+(1-\beta)\overline{E}(h-\alpha)=\beta\underline{E}(f)+(1-\beta)\overline{E}(f)$. Therefore, α must be equal to $\beta(\underline{E}(h)-\underline{E}(f))+(1-\beta)(\overline{E}(h)-\overline{E}(f))$.

In the case that $\alpha<\alpha^*$, we find that $\beta\underline{E}(h-\alpha)+(1-\beta)\overline{E}(h-\alpha)>\beta\underline{E}(f)+(1-\beta)\overline{E}(f)$. So, $h-\alpha$ is Hurwicz but f is not Hurwicz. On the other hand, if $\alpha>\alpha^*$, then $\beta\underline{E}(h-\alpha)+(1-\beta)\overline{E}(h-\alpha)<\beta\underline{E}(f)+(1-\beta)\overline{E}(f)$. Hence, $h-\alpha$ is not Hurwicz but f is still Hurwicz. ■

Using theorem 5, we construct an algorithm for randomly generating a set of k gambles with a precisely specified number of b Hurwicz gambles; see algorithm 4. The process behind algorithm 4 is straightforward, namely, it first generates b Hurwicz gambles and then generates $k-b$ further gambles that are not Hurwicz. Note that in stage 3, to generate a non Hurwicz gamble $f_i-\alpha$ that is harder to detect, we can assign a value of α that is only slightly larger than $\beta(\underline{E}(f_i)-\underline{E}(f_1))+(1-\beta)(\overline{E}(f_i)-\overline{E}(f_1))$. In our simulation, we sample ε from $(0,1)$ and assign $\alpha=\beta(\underline{E}(f_i)-\underline{E}(f_1))+(1-\beta)(\overline{E}(f_i)-\overline{E}(f_1))+\varepsilon$. Note that regardless of how many Hurwicz gambles in a set of k gambles that we want to generate, algorithm 4 needs to evaluate $2k$ natural extensions.

Algorithm 4: Generate a set of k gambles $\mathscr{K}$ such that $\mathrm{opt}_\beta(\mathscr{K})=b\leq k$

Result: a set of k gambles $\mathscr{K}$ such that exactly b gambles are Hurwicz

```
K ← {f_1}; // f_1 is immediately Hurwicz
for i = 2 : b do
    // Generating b − 1 Hurwicz gambles
    for ω ∈ Ω do
        sample f_i(ω) from [0,1]
    end
    α = β(E̲(f_i) − E̲(f_1)) + (1 − β)(Ē(f_i) − Ē(f_1));
    K ← K ∪ {f_i − α}
end
for i = b + 1 : k do
    // Generating k − b non-Hurwicz gambles
    for ω ∈ Ω do
        sample f_i(ω) from [0,1]
    end
    sample α > β(E̲(f_i) − E̲(f_1)) + (1 − β)(Ē(f_i) − Ē(f_1));
    K ← K ∪ {f_i − α}
end
return K
```

Given different sizes of the set $\mathscr{K}$, we consider a range of several options of b that satisfy $b\leq k$ as shown in table 1. Options a to i represents different increasing sizes of Hurwicz gambles in the sets $\mathscr{K}$.

Options	a	b	c	d	e	f	g	h	i
$\lvert\mathscr{K}\rvert=2^4$	1	2	4	6	8	10	12	14	16
$\lvert\mathscr{K}\rvert=2^6$	1	2	4	8	16	20	32	42	64
$\lvert\mathscr{K}\rvert=2^8$	1	2	4	8	16	32	64	128	256

Table 1: Table of options that indicate different sizes of set $\mathscr{K}$ with vary the number of Hurwicz gambles b in $\mathscr{K}$

4.2. Simulation Results

In our simulation, we benchmark our algorithms for Hurwicz (algorithms 1, 2 and 3) on randomly generated sets of gambles that have a precise number of Hurwicz gambles by using algorithm 4. We consider $|\Omega|\in\{2^4,2^6\}$, $|\mathscr{K}|\in\{2^4,2^6,2^8\}$, and $|\mathrm{dom}\,\underline{P}|\in\{2^4,2^6\}$.

To do so, we first generate a lower prevision $\underline{P}$ which avoids sure loss as follows. We first use [13, algorithm 2] with 2^4 coherent previsions to generate a lower prevision $\underline{E}$ on the set of all gambles, that avoids sure loss. With a given finite size of domain, we use [13, stages 1 and 2 in algorithm 4] to restrict $\underline{E}$ to a lower prevision $\underline{P}$ that avoids sure loss, with a specified value for $|\mathrm{dom}\,\underline{P}|$.

To generate gambles $f_i\in\mathscr{K}$, for each ω and i, we sample $f_i(\omega)$ uniformly from the interval $[0,1]$. Next, we use algorithm 4 with $\beta=0.5$ to randomly generate a set $\mathscr{K}$ for $|\mathscr{K}|=k\in\{2^4,2^6,2^8\}$ and $|\mathrm{opt}_\beta(\mathscr{K})|=b$ for some $b\leq k$, using the $\underline{P}$ that we generated earlier for evaluating $\underline{E}$ and $\overline{E}$.

We apply algorithms 1, 2 and 3 on each generated set of gambles $\mathscr{K}$ and measure the total computational time. To evaluate natural extensions inside algorithm 1, we solve linear programs by the standard primal-dual method while for algorithms 2 and 3, we solve the linear programs by the improved primal-dual method which has all improvements from section 3.2. To guarantee a fair comparison, we wrote our own implementation for these algorithms in MATLAB for solving the linear programs. We used the built-in quicksort function to sort all gambles in algorithm 2. The total computational time spent in algorithm 2 includes the time to find an initial feasible p and the time to sort gambles with respect to their expectations. We repeat the process 500 times for each case and show results in fig. 3. All simulations were run on an Intel(R) Core(TM) i3-6100 CPU @ 3.70 GHz processor with 4 GB of RAM.

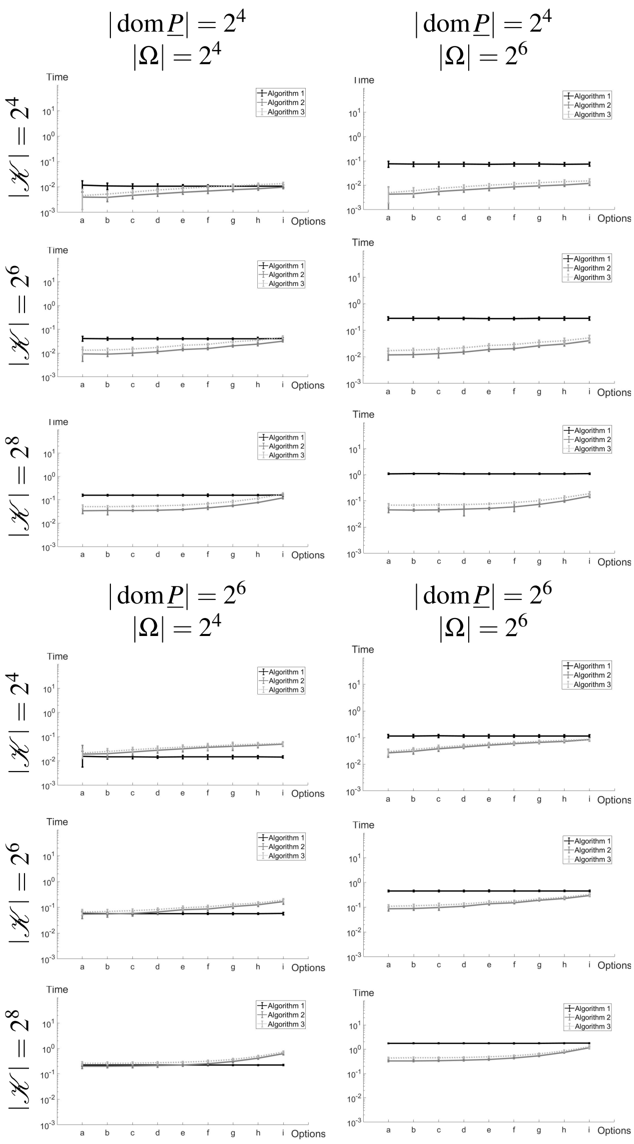

Figure 3: Comparison plots of the average computational time (in seconds) for different algorithms (see the labels for algorithms 1, 2 and 3) for finding Hurwicz gambles where $\beta = 0.5$, $|\operatorname{dom}\underline{P}| = 2^4$ and 2^6. The bars indicate 2 sample standard deviations above and below the sample average, to give an idea of the variability of the computational time. Note that the 95% confidence intervals on the mean computational time are too small to be indicated. The number of outcomes in left and right columns are 2^4 and 2^6. Each row labels a different number of gambles with varying options of the numbers of Hurwicz gambles in the set (see table 1 for each option).

5. Discussion and Conclusion

To summarise, we studied and discussed new algorithms for Hurwicz which is a criterion for decision making with lower previsions. Based on several improvements proposed in [14, 15], we presented how to improve the naive algorithm for Hurwicz and proposed two new algorithms (algorithms 2 and 3) for Hurwicz. Both algorithms 2 and 3 exploit a quick way to compute feasible starting points for primal-dual method to compute lower and upper bound for lower previsions. Providing a sequence of bounds for lower and upper previsions, we also identified ways to eliminate non-Hurwicz gambles early on. The main difference between these algorithms is in the output of the algorithms. Specifically, algorithm 2 will return a single Hurwicz gamble while algorithm 3 will return all Hurwicz gambles in the set.

Moreover, algorithm 3 can be parallelized, whilst algorithm 2 is inherently serial as it must process the gambles in a specific order. However, there are trade-offs between being precise and the number of iterations. In particular, algorithm 3 may need more iterations than algorithm 2 due to comparing intervals with intervals (less precise), whilst algorithm 2 performs a comparison of a single value against intervals (more precise). Note that, however, we designed our implementations and benchmarking on the serial running time to return the overall computational work. A study of parallel algorithms for which these trade-offs can be compared to serial algorithms could be left for future work.

For comparing the performance, we additionally provide a new algorithm for randomly generating artificial sets of decision problems with a pre-determined number of Hurwicz gambles.

Results from our simulation show that the average computational time of three algorithms for Hurwicz depend on the number of outcomes, the number of gambles in the set and the size of the domain of lower previsions. Specifically, if one of these factors is larger, then the mean running time spent on the algorithm is longer. The number of Hurwicz gambles in the set has an impact on the total running time for algorithms 2 and 3 but not for algorithm 1. In particular, the total running time for algorithms 2 and 3 is slightly longer if the number of Hurwicz gambles in the set is increasing while the total running time for algorithm 1 remains the same regardless of the number of Hurwicz gambles.

In practice, for generic decision problems, we normally expect that there is only a single Hurwicz gamble in the set. In such case, we should apply algorithm 2 since it outperforms other algorithms in all scenarios.

If we believe that the set of gambles might contain many Hurwicz gambles, and we are interested in finding only one Hurwicz gamble, then we should apply algorithm 2 unless the size of the domain of lower previsions is large and the numbers of gambles and outcomes is small (see the case

$|\operatorname{dom}\underline{P}| = 2^6$, $|\mathscr{K}| = 2^4$ and $|\Omega| = 2^4$ when there are many Hurwicz gambles) for which algorithm 1 performs better. However, if we are interested in obtaining all Hurwicz gambles in the set, then from the algorithms presented, only algorithm 3 will do.

Finally, we note that all our simulations used $\beta = 0.5$, representing a balanced decision maker. It could also be interesting to investigate different choices of β.

Acknowledgments

This work is supported by Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai, Thailand. We thank the reviewers for their constructive comments.

References

[1] Francis J. Anscombe and Robert J. Aumann. A definition of subjective probability. *Annals of Mathematical Statistics*, 34(1):199–205, March 1963.

[2] Shu-Cherng Fang and Sarat Puthenpura. *Linear Optimization and Extensions: Theory and Algorithms*. Springer Science+Business Media New York, 1993.

[3] Itzhak Gilboa and David Schmeidler. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, 1989. ISSN 0304-4068. doi: 10.1016/0304-4068(89)90018-9.

[4] Nathan Huntley, Robert Hable, and Matthias C. M. Troffaes. *Introduction to Imprecise Probabilities*, chapter Decision making, pages 190–206. Wiley, 2014. doi: 10.1002/9781118763117.ch8.

[5] Christoph Jansen, Thomas Augustin, and Georg Schollmeyer. Decision theory meets linear optimization beyond computation. In Alessandro Antonucci, Laurence Cholvy, and Odile Papini, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 329–339, Cham, 2017. Springer International Publishing. ISBN 978-3-319-61581-3.

[6] Daniel Kikuti, Fabio G. Cozman, and Ricardo S. Filho. Sequential decision making with partially ordered preferences. *Artificial Intelligence*, 175(7): 1346–1365, 2011. doi: 10.1016/j.artint.2010.11.017. Representing, Processing, and Learning Preferences: Theoretical and Practical Challenges.

[7] Isaac Levi. *The Enterprise of Knowledge. An Essay on Knowledge, Credal Probability, and Chance*. MIT Press, Cambridge, 1980.

[8] Paul-Amaury Matt. Uses and computation of imprecise probabilities from statistical data and expert arguments. *International Journal of Approximate Reasoning*, 81:63–86, 2017. doi: 10.1016/j.ijar.2016.11.003.

[9] Enrique Miranda. A survey of the theory of coherent lower previsions. *International Journal of Approximate Reasoning*, 48(2):628–658, 2008. doi: 10.1016/j.ijar.2007.12.001.

[10] Enrique Miranda and Gert de Cooman. *Introduction to Imprecise Probabilities*, chapter Lower prevision, pages 28–55. Wiley, 2014. doi: 10.1002/9781118763117.ch2.

[11] Nawapon Nakharutai. Algorithms for generating sets of gambles for decision making with lower previsions. In Van-Nam Huynh, Tomoe Entani, Chawalit Jeenanunta, Masahiro Inuiguchi, and Pisal Yenradee, editors, *Integrated Uncertainty in Knowledge Modelling and Decision Making*, pages 62–71, Cham, 2020. Springer International Publishing.

[12] Nawapon Nakharutai, Matthias C.M. Troffaes, and Camila C.S. Caiado. Efficient algorithms for checking avoiding sure loss. *Proceedings of Machine Learning Research*, 62:241–252, 2017.

[13] Nawapon Nakharutai, Matthias C.M. Troffaes, and Camila C.S. Caiado. Improved linear programming methods for checking avoiding sure loss. *International Journal of Approximate Reasoning*, 101:293–310, October 2018. doi: 10.1016/j.ijar.2018.07.013.

[14] Nawapon Nakharutai, Matthias C.M. Troffaes, and Camila C.S. Caiado. Improving and benchmarking of algorithms for decision making with lower previsions. *International Journal of Approximate Reasoning*, 113: 91–105, July 2019. doi: 10.1016/j.ijar.2019.06.008.

[15] Nawapon Nakharutai, Matthias C.M. Troffaes, and Camila C.S. Caiado. Improving and benchmarking of algorithms for Γ-maximin, Γ-maximax and interval dominance. *International Journal of Approximate Reasoning*, 133:95–115, June 2021. doi: 10.1016/j.ijar.2021.03.005.

[16] Teddy Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: Gamma-maximin versus E-admissibility. *Synthese*, 140(1-2):69–88, May 2004.

[17] Matthias C. M. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45(1):17–29, may 2007. doi: 10.1016/j.ijar.2006.06.001.

[18] Matthias C. M. Troffaes and Gert de Cooman. *Lower Previsions*. Wiley Series in Probability and Statistics. Wiley, 2014. ISBN 978-0-470-72377-7. URL http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470723777.html.

[19] Matthias C. M. Troffaes and Robert Hable. *Introduction to Imprecise Probabilities*, chapter Computation, pages 329–337. Wiley, 2014. doi: 10.1002/9781118763117.ch16.

[20] Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[21] Peter M. Williams. Notes on conditional previsions. Technical report, School of Math. and Phys. Sci., Univ. of Sussex, 1975.

[22] Peter M. Williams. Notes on conditional previsions. *International Journal of Approximate Reasoning*, 44 (3):366–383, 2007. doi: 10.1016/j.ijar.2006.07.019.

[23] Marco Zaffalon, Keith Wesnes, and Orlando Petrini. Reliable diagnoses of dementia by the naive credal classifier inferred from incomplete cognitive data. *Artificial Intelligence in Medicine*, 29(1):61–79, 2003. ISSN 0933-3657. doi: 10.1016/S0933-3657(03)00046-0. Artificial Intelligence in Medicine Europe AIME '01.