IMPROVED LINEAR PROGRAMMING METHODS FOR CHECKING AVOIDING SURE LOSS

NAWAPON NAKHARUTAI, MATTHIAS C. M. TROFFAES, AND CAMILA C. S. CAIADO

ABSTRACT. We review the simplex method and two interior-point methods (the affine scaling and the primal-dual) for solving linear programming problems for checking avoiding sure loss, and propose novel improvements. We exploit the structure of these problems to reduce their size. We also present an extra stopping criterion, and direct ways to calculate feasible starting points in almost all cases. For benchmarking, we present algorithms for generating random sets of desirable gambles that either avoid or do not avoid sure loss. We test our improvements on these linear programming methods by measuring the computational time on these generated sets. We assess the relative performance of the three methods as a function of the number of desirable gambles and the number of outcomes. Overall, the affine scaling and primal-dual methods benefit from the improvements, and they both outperform the simplex method in most scenarios. We conclude that the simplex method is not a good choice for checking avoiding sure loss. If problems are small, then there is no tangible difference in performance between all methods. For large problems, our improved primal-dual method performs at least three times faster than any of the other methods.

1. INTRODUCTION

In statistical modelling, we often face issues such as limited structural information about dependencies, lack of data, limited expert opinion, or even contradicting information from different experts. Various authors [16, 17, 14, 11] have argued that these issues can be handled by modelling our beliefs using *sets of desirable gambles*. A gamble represents a reward (e.g. monetary) that depends on an uncertain outcome. We can model our beliefs about this outcome by stating a collection of gambles that we are willing to accept. Such set is called a set of desirable gambles. Through duality, every set of desirable gambles is mathematically equivalent to a set of probability distributions.

If there are no combinations of desirable gambles that result in a certain loss, then we say that our set of desirable gambles *avoids sure loss* [16, 17]. To verify whether a set of desirable gambles avoids sure loss, we can solve a linear programming problem [14, p. 151].

Linear programs for checking avoiding sure loss have been studied for instance in [15, 9]. However, these studies focus on forming linear programs and do not mention which algorithms we should use. In the early '90s, Walley [14, p. 551] mentioned that Karmarkar's method can be considered for solving large linear programs. However, nowadays Karmarkar's method is considered obsolete in favour of other interior point methods such as affine scaling and primal-dual methods [1].

The simplex method is one of the oldest and simplest methods, and the affine scaling method is an improved version of Karmarkar's method, whilst the primal-dual method is currently considered one of the best general purpose methods. In previous work, we presented an initial comparative study of these three methods for checking avoiding sure loss [8]. In that study, we slightly reduced the size of the problems and proposed two improvements: an extra stopping criterion to detect unboundedness more quickly, and a direct way to calculate feasible starting points. There, we also quantified the impact of these improvements [8, Fig. 1], for the primal-dual method.

In this paper, our main goal is to elaborate on the improvements in [8], and to further develop efficient algorithms for checking avoiding sure loss. In particular, we study also the dual of each linear program, and we generalise the process to find feasible starting points. We also discuss in detail the advantages and disadvantages of each method for checking avoiding sure loss. For benchmarking, we provide a variety of algorithms for generating sets of desirable gambles. In a simulation study, we generate random sets of desirable gambles and assess the impact of our improvements. In addition, we provide proofs for all relevant results, including some that were stated without proof in [8].

The paper is organised as follows. Section 2 gives a brief outline of avoiding sure loss and coherence. Section 3 studies several linear programming problems for checking avoiding sure loss, and we slightly reduce the size of these linear programming problems. Section 4 reviews the simplex, the affine scaling and the primal-dual algorithms, and studies how we can improve these algorithms to check avoiding sure loss. Sections 5 and 6 present some algorithms for generating random sets of desirable gambles. Section 7 compares the efficiency of our improved methods. Section 8 concludes the paper.

2. Desirable gambles and lower previsions

In this section, we explain desirable gambles, lower previsions, avoiding sure loss, coherence, and natural extension [14]. We also introduce the notation used throughout.

2.1. **Avoiding sure loss.** Let Ω be a finite set of uncertain outcomes. A *gamble* is a bounded real-valued function on Ω. Let $\mathcal{L}(\Omega)$ denote the set of all gambles on Ω. Let $\mathcal{D}$ be a finite set of gambles that a subject decides to accept; we call $\mathcal{D}$ the subject's *set of desirable gambles*. The desirability axioms essentially state that a non-negative combination of desirable gambles should not produce a sure loss [14, §2.3.3]. In that case, we say that $\mathcal{D}$ avoids sure loss.

Definition 1. *[14, §3.7.1] A set $\mathcal{D} \subseteq \mathcal{L}(\Omega)$ is said to* avoid sure loss *if for all $n \in \mathbb{N}$, all $\lambda_1, \ldots, \lambda_n \geq 0$, and all $f_1, \ldots, f_n \in \mathcal{D}$,*

$$(1) \qquad \max_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) \right) \geq 0.$$

We can also model uncertainty via acceptable buying (or selling) prices for gambles. A *lower prevision* $\underline{P}$ is a real-valued function defined on some subset of $\mathcal{L}(\Omega)$. We denote the domain of $\underline{P}$ by $\operatorname{dom} \underline{P}$. Given a gamble $f \in \operatorname{dom} \underline{P}$, we interpret $\underline{P}(f)$ as a subject's supremum buying price for f.

Definition 2. *[14, §2.4.2] A lower prevision $\underline{P}$ is said to* avoid sure loss *if for all $n \in \mathbb{N}$, all $\lambda_1, \ldots, \lambda_n \geq 0$, and all $f_1, \ldots, f_n \in \operatorname{dom} \underline{P}$,*

$$\text{(2)} \qquad \max_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i \left[f_i(\omega) - \underline{P}(f_i) \right] \right) \geq 0.$$

Any lower prevision $\underline{P}$ induces a *conjugate upper prevision* $\overline{P}$ on $-\operatorname{dom} \underline{P} := \{ -f \colon f \in \operatorname{dom} \underline{P} \}$, defined by $\overline{P}(f) := -\underline{P}(-f)$ for all $f \in -\operatorname{dom} \underline{P}$ [14, §2.3.5]. $\overline{P}(f)$ represents a subject's infimum selling price for f. $\underline{P}$ is said to be *self-conjugate* when $\operatorname{dom} \underline{P} = -\operatorname{dom} \underline{P}$ and $\underline{P}(f) = \overline{P}(f)$ for all $f \in \operatorname{dom} P$. We simply call a self-conjugate lower prevision $\underline{P}$ a *prevision* and write it as P [11, p. 41].

2.2. **Coherence.** Coherence is another rationality condition for lower previsions and is stronger than avoiding sure loss. Coherence requires that the subject's supremum buying prices for gambles cannot be increased by considering any finite non-negative linear combination of other desirable gambles [14, §2.5.2]. In section 5, we will use coherent lower previsions to generate sets of desirable gambles that avoids sure loss.

Definition 3. *[14, §2.5.4] A lower prevision $\underline{P}$ is said to be* coherent *if for all $n \in \mathbb{N}$, all $\lambda_0, \ldots, \lambda_n \geq 0$ and all $f_0, \ldots, f_n \in \operatorname{dom} \underline{P}$,*

$$\text{(3)} \qquad \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i [f_i(\omega) - \underline{P}(f_i)] - \lambda_0 [f_0(\omega) - \underline{P}(f_0)] \right) \geq 0.$$

Next, we give some examples of coherent lower previsions. The lower prevision given by $\underline{P}(f) := \inf f$ for all $f \in \mathcal{L}(\Omega)$ is coherent, and is called the *vacuous* lower prevision [14, §2.3.7]. Previsions that avoid sure loss are also coherent:

Theorem 1. *[14, p.87] A prevision P is coherent if and only if it avoids sure loss (as a lower prevision).*

The expectation of f associated with the probability mass function p is given by

$$\text{(4)} \qquad E_p(f) := \sum_{\omega \in \Omega} p(\omega) f(\omega).$$

An expectation operator is coherent as well.

Generating probability mass functions is easy (see algorithm 1 further), and we can use them to generate other coherent lower previsions via lower envelopes and convex combinations:

Definition 4. *[11, p. 60] Let Γ be a non-empty collection of lower previsions defined on a common domain $\mathcal{K}$. A lower prevision $\underline{Q}$ is called the* lower envelope *of Γ if*

$$\text{(5)} \qquad \underline{Q}(f) = \inf_{\underline{P} \in \Gamma} \underline{P}(f) \text{ for all } f \in \mathcal{K}.$$

Theorem 2. *[11, p. 61] If all lower previsions in Γ are coherent, then the lower envelope of Γ is also coherent.*

We define the unit simplex as the set of all probability mass functions:

$$\text{(6)} \qquad \Delta(\Omega) := \left\{ p \in \mathbb{R}^{\Omega} \colon p \geq 0 \text{ and } \sum_{\omega \in \Omega} p(\omega) = 1 \right\}.$$

Its extreme points are the $\{0, 1\}$-valued probability mass functions [14, §3.2.6]. The *credal set* of a lower prevision $\underline{P}$ is defined by

$$(7) \qquad \mathcal{M}_{\underline{P}} := \{p \in \Delta(\Omega) : \forall f \in \operatorname{dom} \underline{P}, \ E_p(f) \geq \underline{P}(f)\}.$$

$\mathcal{M}_{\underline{P}}$ completely determines $\underline{P}$ if $\underline{P}$ is coherent and there is a one-to-one correspondence between coherent lower previsions on $\mathcal{L}(\Omega)$ and closed convex subsets of $\Delta(\Omega)$ [11, p. 79]. Moreover, it suffices to consider the set of extreme points $\operatorname{ext} \mathcal{M}_{\underline{P}}$ of $\mathcal{M}_{\underline{P}}$ [14, p. 145]:

Theorem 3. *(adapted from [14, p. 146]) Let $\underline{P}$ be a coherent lower prevision. Then for every $f \in \operatorname{dom} \underline{P}$, there is a $p \in \operatorname{ext} \mathcal{M}_{\underline{P}}$ such that $\underline{P}(f) = E_p(f)$.*

When $\operatorname{ext} \mathcal{M}_{\underline{P}}$ is finite, then $\underline{P}$ is called *polyhedral*. We can construct a polyhedral lower prevision as follows. Let M be a finite set of probability mass functions on Ω. A polyhedral lower prevision is then given by [2, §9.2.1]

$$(8) \qquad \underline{E}_M(f) := \min_{p \in M} E_p(f).$$

Theorem 4. *[14, p. 79] Let $\underline{P}_1$ and $\underline{P}_2$ be lower previsions on the same domain. Let $0 \leq \delta \leq 1$. If $\underline{P}_1$ and $\underline{P}_2$ are coherent, then so is $(1 - \delta)\underline{P}_1 + \delta\underline{P}_2$.*

Let P_0 be a coherent prevision on $\mathcal{L}(\Omega)$ and let $0 \leq \delta \leq 1$. The lower prevision defined on all $f \in \mathcal{L}(\Omega)$ by

$$(9) \qquad \underline{P}(f) := (1 - \delta)P_0(f) + \delta \inf f$$

is called a *linear-vacuous mixture* [14, §2.9.2], and is coherent by theorem 4.

2.3. **Natural extension.** The natural extension of a set of desirable gambles $\mathcal{D}$ is defined as the smallest set of gambles which includes all finite non-negative combinations of gambles in $\mathcal{D}$ and all non-negative gambles [11, § 3.7]:

Definition 5. *[11, p. 32] The natural extension of a set $\mathcal{D} \subseteq \mathcal{L}(\Omega)$ is:*

$$(10) \qquad \mathcal{E}_{\mathcal{D}} := \left\{ g_0 + \sum_{i=1}^n \lambda_i g_i : g_0 \geq 0, \ n \in \mathbb{N}, \ g_1, \ldots, g_n \in \mathcal{D}, \ \lambda_1, \ldots, \lambda_n \geq 0 \right\}.$$

From this natural extension, we can derive a supremum buying price for any gamble f:

Definition 6. *For any set $\mathcal{D} \subseteq \mathcal{L}(\Omega)$ and $f \in \mathcal{L}(\Omega)$, we define:*

$$\underline{E}_{\mathcal{D}}(f) := \sup\{\alpha \in \mathbb{R} : f - \alpha \in \mathcal{E}_{\mathcal{D}}\}$$

$$(11) \qquad = \sup\left\{\alpha \in \mathbb{R} : f - \alpha \geq \sum_{i=1}^n \lambda_i f_i, n \in \mathbb{N}, f_i \in \mathcal{D}, \lambda_i \geq 0\right\}.$$

Note that we can derive a set of desirable gambles from $\underline{P}$ as follows [11, p. 42]:

$$(12) \qquad \mathcal{D}_{\underline{P}} := \{g - \mu : g \in \operatorname{dom} \underline{P} \text{ and } \mu < \underline{P}(g)\}.$$

Putting this all together, we can define the natural extension of $\underline{P}$:

Definition 7. *[14, §3.1.1] Let $\underline{P}$ be a lower prevision. The natural extension of $\underline{P}$ is a lower prevision defined on all $f \in \mathcal{L}(\Omega)$ and given by:*

$$(13) \quad \underline{E}_{\underline{P}}(f) := \underline{E}_{\mathcal{D}_{\underline{P}}}(f)$$

$$= \sup\left\{\alpha \in \mathbb{R} \colon f - \alpha \geq \sum_{i=1}^{n} \lambda_i(f_i - \underline{P}(f_i)), n \in \mathbb{N}, \, f_i \in \mathrm{dom}\,\underline{P}, \, \lambda_i \geq 0\right\}.$$

3. Linear programming

In this section, we briefly review linear programming, and we study several linear programming problems for checking avoiding sure loss.

3.1. Linear programming problems.

Any problem of minimising or maximising a linear function, called the objective function, subject to linear constraints, is called a linear program, and can always be written in the form of eq. (P), with dual given in eq. (D):

(P) $\qquad\qquad \min c^{\mathsf{T}} x$ subject to $Ax = b, \; x \geq 0$

(D) $\qquad\qquad \max b^{\mathsf{T}} y$ subject to $A^{\mathsf{T}} y + t = c, \; t \geq 0, \; y$ free

where $A \in \mathbb{R}^{m \times n}$ has rank m, $m \leq n$, and x, c, t, b, and y are vectors with dimensions as expected (so x, c, $t \in \mathbb{R}^n$ and b, $y \in \mathbb{R}^m$). We call eq. (P) the *primal problem* and eq. (D) the *dual problem*. They have the same solution [4, p. 59], so we can solve either one of them.

A solution is called *feasible* if it satisfies all constraints. The primal problem is called *unbounded* if for all $\lambda \in \mathbb{R}$, there is a feasible solution x such that $c^{\mathsf{T}} x \leq \lambda$. A feasible solution that achieves the optimal value of the objective function is called an *optimal* solution. A *basis* is a collection of m of the n variables that correspond to m linearly independent columns of A. A variable in the basis is called a *basic* variable; otherwise it is called a *non-basic* variable. A *basic* feasible solution is a feasible solution such that all of the $n - m$ non-basic variables are zero. The basic feasible solutions are precisely the extreme points of the feasible region.

A linear programming problem is called *degenerate* if it has basic feasible solutions with $n - m + 1$ or more zero elements [4, p. 22]. If all basic feasible solutions are zero (this happens when $b = 0$), we say that the problem is *fully* degenerate. In this case, the feasible region is a cone, and therefore has only one extreme point, namely, the origin. As we will see later, one way to check avoiding sure loss is to solve a fully degenerate problem. The following lemma is useful for finding the optimal value of a fully degenerate problem.

Lemma 1. *(A generalised version of [13, p. 42, exercise 3.4].) The linear programming problem*

$$(14) \qquad\qquad\qquad \min \quad c^{\mathsf{T}} x$$

$$(15) \qquad\qquad\qquad subject\ to \quad Ax \geq 0$$

either has optimal value equal to zero, or is unbounded below.

Lemma 1 shows that, for fully degenerate problems, if there is a feasible solution x such that $c^{\mathsf{T}} x < 0$, then the problem is unbounded. Therefore, in our algorithms, we can stop early as soon as we find a negative value.

3.2. **Linear programming for checking avoiding sure loss.** We now present some linear programming problems for checking avoiding sure loss. The problems (P1) and (D1) in theorem 5 are similar to the linear programming problems discussed for lower previsions in **(author?)** [14, p. 175].

Theorem 5. *The set $\mathcal{D} = \{f_1, \ldots, f_n\}$ avoids sure loss if and only if the optimal value of* (P1) *is zero:*

(P1a)   (P1) $\min \quad \alpha$

(P1b) $\text{subject to} \quad \forall \omega \in \Omega \colon \sum_{i=1}^{n} f_i(\omega) \lambda_i - \alpha \leq 0$

(P1c) $\forall i \colon \lambda_i \geq 0 \quad (\alpha \text{ free}),$

or, equivalently, if and only if its dual problem, (D1), *has a feasible solution:*

(D1a)   (D1) $\max \quad 0$

(D1b) $\text{subject to} \quad \forall f_i \in \mathcal{D} \colon \sum_{\omega \in \Omega} f_i(\omega) p(\omega) \geq 0$

(D1c) $\sum_{\omega \in \Omega} p(\omega) = 1$

(D1d) $\forall \omega \colon p(\omega) \geq 0.$

Note that (P1) is fully degenerate, and (D1) is nearly fully degenerate. Clearly, any feasible solution of (D1) is also an optimal solution, since the objective function is constant.

3.3. **Reduced linear programming problem for checking avoiding sure loss.** When solving linear programs, several algorithms, such as the simplex and the affine scaling methods, require all variables to be non-negative. Here we present alternative linear programming problems which are slightly smaller in dimension and have only non-negative variables. The following theorem is presented in [8] without a proof; we give a proof in the appendix.

Theorem 6. *[8] Choose any $\omega_0 \in \Omega$. The set $\mathcal{D} = \{f_1, \ldots, f_n\}$ avoids sure loss if and only if the optimal value of* (P2) *is zero:*

(P2a)   (P2) $\min \quad \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha$

(P2b) $\text{subject to} \quad \forall \omega \neq \omega_0 \colon \sum_{i=1}^{n} (f_i(\omega_0) - f_i(\omega)) \lambda_i + \alpha \geq 0$

(P2c) $\forall i \colon \lambda_i \geq 0 \ \text{and} \ \alpha \geq 0,$

or, equivalently, if and only if its dual problem, (D2), *has a feasible solution:*

(D2a)   (D2) $\max \quad 0$

(D2b) $\text{subject to} \quad \forall f_i \in \mathcal{D} \colon \sum_{\omega \neq \omega_0} (f_i(\omega_0) - f_i(\omega)) p(\omega) \leq f_i(\omega_0)$

(D2c) $\sum_{\omega \neq \omega_0} p(\omega) \leq 1$

(D2d) $\forall \omega \colon p(\omega) \geq 0.$

Proof. See appendix. □

(P2) is still fully degenerate, whilst (D2) is no longer degenerate.

As primal optimality corresponds to dual feasibility [5, p. 104], if we choose any ω_0 such that most values $f_i(\omega_0)$ are non-negative, then (D2) will be closer to a dual feasible solution, and therefore (P2) will also be closer to a primal optimal solution. For instance, if there is an ω_0 for which $f_i(\omega_0) \geq 0$ for all i, then we directly obtain a feasible solution of (D2) by setting $p(\omega) = 0$ for all $\omega \neq \omega_0$ [8]. The corresponding optimal solution of (P2) is given by $\lambda_i = 0$ for all i and $\alpha = 0$.

Next, we look at the simplex, the affine scaling and the primal-dual algorithms. We briefly explain how they work, and discuss improvements.

4. Improving algorithms for solving linear programs

4.1. **Simplex methods.** The simplex method is an iterative algorithm that needs an extreme point to start. If a problem can be written in the form:

(S) $$\min c^{\mathsf{T}} x \text{ subject to } Ax + s = b, \ x \geq 0, \ s \geq 0,$$

then provided that $b \geq 0$, we immediately obtain a starting extreme point by setting $s = b$ and $x = 0$ [6, §4.2]. Equation (S) can be represented in a table. The simplex algorithm performs row operations on this table to move between extreme points, improving the value of the objective function at each iteration, until we find the optimal value [6, §4.4].

The problem (P2) can be easily written as eq. (S) by negating eq. (P2b) and adding non-negative slack variables $s(\omega)$ [8]:

(P3a) (P3) $$\min \quad \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha$$

(P3b) $$\text{subject to} \quad \forall \omega \neq \omega_0 \colon \sum_{i=1}^{n} (f_i(\omega) - f_i(\omega_0))\lambda_i - \alpha + s(\omega) = 0$$

(P3c) $$\forall i \colon \lambda_i \geq 0, \ \forall \omega \neq \omega_0 \colon s(\omega) \geq 0 \text{ and } \alpha \geq 0.$$

Setting all λ_i, α and $s(\omega)$ to zero provides an initial extreme point. Because (P3) is fully degenerate, this is also the only extreme point.

Full degeneracy also implies that, by the minimum ratio test [4, p. 36], we can select any leaving basic variable at any point in the simplex method. Moreover, the value of any new entering basic variable is always zero, so the objective value never improves. This is bad news: if the same sequence of degenerate pivots associated with a non-optimal solution is repeated multiple times, then the simplex method is said to be *cycling* and will never terminate [4, §3.5]. Fortunately, there are simple ways to avoid cycling, such as the lexicographic method or Bland's rule [4, §3.6].

Once the problem of cycling is addressed, we may encounter another problem called *stalling*, meaning that the method performs an exponentially long sequence of degenerate pivots [3]. Since there are no known efficient rules for choosing leaving variables for fully degenerate problems, we may end up visiting all extreme points. Efficiently solving fully degenerate problems is still an open problem [7].

We now consider the dual problem (D2) and convert it into the form of eq. (S). We first add non-negative slack variables to eqs. (D2b) and (D2c) and obtain equality constraints. Since we want all the right hand side values to be non-negative, for every j such that $f_j(\omega) < 0$, we multiply the corresponding constraint by -1 to

make it non-negative and then add a non-negative artificial variable. In this case, the size of the linear programming problem is slightly bigger. We arrive at [8]:

(D3a)   (D3)   min $\sum_{j \in N} v_j$

(D3b)   s.t. $\forall j \in N\colon \sum_{\omega \neq \omega_0} (f_j(\omega) - f_j(\omega_0))p(\omega) - s_j + v_j = -f_j(\omega_0)$

(D3c) $\forall j \notin N\colon \sum_{\omega \neq \omega_0} (f_j(\omega_0) - f_j(\omega))p(\omega) + s_j = f_j(\omega_0)$

(D3d) $\sum_{\omega \neq \omega_0} p(\omega) + q = 1$

(D3e) $\forall \omega \neq \omega_0\colon p(\omega) \geq 0,\ \forall j\colon s_j \geq 0,\ v_j \geq 0 \text{ and } q \geq 0$

with $N := \{j \in N\colon f_j(\omega_0) < 0\}$. An initial extreme point for (D3) is given by $v_j = -f_j(\omega_0), s_j = 0$ for all $j \in N$, $s_j = f_j(\omega_0)$ for all $j \notin N$, $p(\omega) = 0$ for all $\omega \neq \omega_0$ and $q = 1$. If all $f_j(\omega_0) \geq 0$, then we have an immediate optimal solution. The problem (D3) is normally non-degenerate, except if $f_j(\omega_0) = 0$ for some j [8].

To summarise, to check avoiding sure loss by the simplex method, we can solve either (P3), which is fully degenerate, or (D3), whose size is slightly larger. Even though the simplex method may stall under degeneracy, in practice, it is still one of the most commonly used algorithms. This is why we treated it here.

4.2. **Affine scaling methods.** The affine scaling method solves linear programs of the form of eq. (P). Given a starting interior feasible point, the method generates a sequence of interior feasible points which iteratively decrease the value of the objective function, until the improvement is small enough or unboundedness is detected [4, §7.1.1].

The affine scaling method can solve (P3) as it is already in the form of eq. (P). Similar to the simplex method, degeneracy can affect the performance of the affine scaling method. **(author?)** [12] show that this can be overcome by limiting the step-size of the algorithm.

However, unlike the simplex method, lemma 1 can be applied to the affine scaling method. Specifically, when the affine scaling solves (P3), the method can stop as soon as it finds a negative value for the objective function.

The dual problem (D2) can be written in the form of eq. (P) by adding non-negative slack variables:

(D4a)   (D4)   max   0

(D4b)   subject to $\forall f_i \in \mathcal{D}\colon \sum_{\omega \neq \omega_0} (f_i(\omega_0) - f_i(\omega))p(\omega) + t_i = f_i(\omega_0)$

(D4c) $\sum_{\omega \neq \omega_0} p(\omega) + q = 1$

(D4d) $\forall \omega \neq \omega_0\colon p(\omega) \geq 0,\ \forall i\colon t_i \geq 0,\ q \geq 0.$

Although we can also solve (D3) by the affine scaling method, we solve (D4) as it has fewer artificial variables.

As the affine scaling method requires an initial interior feasible point, we normally need to solve two linear programming problems: one to find a starting interior feasible point, and another one to solve the original problem with this starting point.

A starting interior feasible solution can be found as follows (see [4, §7.1.2] for more details). Consider the constraints $Ax = b$ and $x \geq 0$. Choose any point $x^0 > 0$ and calculate $z = b - Ax^0$. If $z = 0$, then x^0 is an interior feasible solution of the original problem. Otherwise, solve

(P') $\qquad\qquad$ $\min \gamma$ subject to $Ax + z\gamma = b,\ x \geq 0,\ \gamma \geq 0$

by the affine scaling method, using $[x\ \gamma] = [x^0\ 1]$ as a starting point (this point is an interior feasible solution of eq. (P')). If an optimal solution is $[x^*\ \gamma^*]$ with $\gamma^* = 0$, then x^* is an interior feasible solution of the original problem. Otherwise, there is no feasible solution.

The good news for us is that, for solving either (P3) or (D4), we only need to solve a single linear programming problem. For (D4), this is because every interior feasible point is also an optimal solution, so we only need to solve eq. (P'). For (P3), due to the structure of the problem, we can immediately write down an interior feasible point in closed form, so we do not need to solve eq. (P').

Let us explain how eq. (P') looks like for (D4), following (with a slight improvement here) **(author?)** [8]. Let $\Omega \backslash \{\omega_0\} = \{\omega_1, \dots, \omega_m\}$. Consider, for the moment, an arbitrary

(16) $\qquad\qquad x^0 = \begin{bmatrix} p^0(\omega_1) & \cdots & p^0(\omega_m) & t_1^0 & \cdots & t_n^0 & q^0 \end{bmatrix} > 0$

and define $[r\ z] := b - Ax^0$, so

(17) $\qquad\qquad r_i := h_i - t_i^0$

(18) $\qquad\qquad z := 1 - \left(\sum_{\omega \neq \omega_0} p^0(\omega) + q^0 \right)$

where

(19) $\qquad\qquad h_i := f_i(\omega_0) - \sum_{\omega \neq \omega_0} (f_i(\omega_0) - f_i(\omega)) p^0(\omega).$

If we choose $q^0 = p^0(\omega) = 1/|\Omega|$ for all $\omega \neq \omega_0$, then $z = 0$. Choose $t_i^0 = 1$ (or any other strictly positive value) for all i where $h_i \leq 0$. Finally, choose $t_i^0 = h_i$ for all i where $h_i > 0$, so all corresponding r_i are zero. So, eq. (P') becomes:

(D4'a) $\qquad$ (D4') $\quad \min \quad \gamma$

(D4'b) $\qquad\qquad\qquad$ s.t. $\quad \forall i \colon \sum_{\omega \neq \omega_0} (f_i(\omega_0) - f_i(\omega)) p(\omega) + t_i + r_i \gamma = f_i(\omega_0)$

(D4'c) $\qquad\qquad\qquad\qquad\qquad \sum_{\omega \neq \omega_0} p(\omega) + q = 1$

(D4'd) $\qquad\qquad\qquad\qquad \forall \omega \neq \omega_0 \colon p(\omega) \geq 0,\ \forall i \colon t_i \geq 0,\ q \geq 0 \text{ and } \gamma \geq 0,$

with an initial interior feasible point as constructed. Note that for simplicity in our implementation, we choose $t_i^0 = 1$ for all i. If the optimal solution of (D4') has $\gamma^* = 0$, then we will have found an interior feasible solution for (D4) (and therefore also an optimal solution for (D4)), and so $\mathcal{D}$ avoids sure loss; otherwise, there is no feasible solution and $\mathcal{D}$ incurs sure loss.

For (P3), we simply calculate a starting interior feasible point using theorem 7 below, with $\lambda_i^0 = 1$.

Theorem 7. *[8, modified] An interior feasible point of the following system of linear constraints*

$$(20) \qquad \forall j \in \{1, \ldots, m\} \colon \sum_{i=1}^{n} a_{ij}\lambda_i - \alpha + s_j = b_j$$

$$(21) \qquad \forall i \colon \lambda_i \geq 0, \ \forall j \colon s_j \geq 0, \ \alpha \geq 0$$

is given by setting $\lambda_i = \lambda_i^0$ for some arbitrary $\lambda_i^0 > 0$, $\alpha = 1 + \max\{0, -\delta\}$ with

$$(22) \qquad \delta := \min_j \left\{ b_j - \sum_{i=1}^{n} a_{ij}\lambda_i^0 \right\},$$

and $s_j = b_j - \sum_{i=1}^{n} a_{ij}\lambda_i^0 + \alpha$.

Proof. We must show that eq. (20) is satisfied, and that all variables are strictly positive.

Clearly, eq. (20) is satisfied by our choice of s_j, all $\lambda_i = \lambda_i^0 > 0$, and $\alpha \geq 1 > 0$. Finally, note that also all $s_j > 0$ because

$$(23) \qquad s_j = b_j - \sum_{i=1}^{n} a_{ij}\lambda_i^0 + \alpha \geq \delta + \alpha \geq \delta + 1 - \delta > 0,$$

where we used the definitions of δ and α respectively. □

To conclude, to check avoiding sure loss with the affine scaling method, we either solve (P3) or (D4'). In either case, we have a closed form initial interior feasible point. In addition, we can apply lemma 1 as an extra stopping rule to detect unboundedness when solving (P3). However, we need to take care to limit the step-size due to degeneracy.

Next, we look at another interior-point method for which we can also apply lemma 1 and theorem 7, but which does not have a limitation on the step-size and which has faster convergence as observed in practice.

4.3. **Primal-dual methods.** The primal-dual method is an iterative algorithm which finds an optimal solution by solving eqs. (P) and (D) simultaneously. At each iteration, the method solves the following system:

$$(24) \qquad \begin{bmatrix} Ax - b \\ A^{\mathsf{T}}y + t - c \\ x^{\mathsf{T}}t \end{bmatrix} = 0, \ x \geq 0, \ t \geq 0 \qquad (y \text{ free})$$

whilst keeping the variables x and t positive. Theoretically, given an initial interior feasible point (x, y, t) where $x > 0$ and $t > 0$, the method can generate a sequence of interior feasible points such that $x^{\mathsf{T}}t$ gets closer and closer to zero (by duality, the solution is optional when $x^{\mathsf{T}}t = 0$; the method simply exploits this fact). However, in practical implementations, keeping (x, y, t) in the feasible region is very difficult due to numerical issues [4, §7.3].

Therefore, in practical implementations, the primal-dual method starts with an arbitrary point (x, y, t) where $x > 0$ and $t > 0$ and generates points that converge to a feasible optimal solution. The method will stop when the primal residual $Ax - b$, dual residual $A^{\mathsf{T}}y + t - c$, and duality gap $x^{\mathsf{T}}t$ are small enough, or when unboundedness in either the primal or the dual is detected. Although this modified

version of the algorithm has no known convergence proof, it works extremely well in practice [4, §7.3].

The problem (P3) is already in the form of eq. (P). Its dual is [8]:

(D5a) $\quad$ (D5) $\quad$ max $\quad 0$

(D5b) $\qquad$ s. t. $\quad \forall f_i \in \mathcal{D}:\ \sum_{\omega \neq \omega_0} (f_i(\omega) - f_i(\omega_0))v(\omega) + t_i = f_i(\omega_0)$

(D5c) $\qquad\qquad q - \sum_{\omega \neq \omega_0} v(\omega) = 1$

(D5d) $\qquad\qquad \forall \omega \neq \omega_0:\ v(\omega) + p(\omega) = 0$

(D5e) $\qquad\qquad \forall i:\ t_i \geq 0,\ \forall \omega \neq \omega_0:\ p(\omega) \geq 0$ and $q \geq 0$.

Because $-v(\omega) = p(\omega) \geq 0$, (D5) is equivalent to (D4), as expected. The primal-dual method solves (P3) and (D5) simultaneously.

Theorem 7 provides an initial interior feasible point for (P3). However, there is no closed form feasible point for (D5) (if we had, then we immediately would have found an optimal solution). In this case, a starting point of (D5) can be $q^0 = p^0(\omega) = 1/|\Omega|$, $v^0(\omega) = -1/|\Omega|$, and $t_i^0 = 1$ for all i [8].

Remind that we can apply lemma 1 to (P3) only if we can keep all iterative points in the feasible region. Although we start (P3) with a feasible point, the next points do not necessarily remain in the feasible region due to numerical rounding errors. Therefore, it is good practice to calculate the primal residual and only apply lemma 1 if this error is neglegible [8].

Now, consider solving the problem (D4') by the primal-dual method. In this case, the dual of (D4') can be written in the form of eq. (D) as follows:

(P4'a) $\quad$ (P4') $\quad$ max $\quad \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha$

(P4'b) $\qquad$ s. t. $\quad \forall \omega \neq \omega_0:\ \sum_{i=1}^{n}(f_i(\omega_0) - f_i(\omega))\lambda_i + \alpha + s(\omega) = 0$

(P4'c) $\qquad\qquad \forall i:\ \lambda_i + u_i = 0$

(P4'd) $\qquad\qquad \alpha + \beta = 0$

(P4'e) $\qquad\qquad \sum_{i=1}^{n} r_i \lambda_i + \mu = 1$

(P4'f) $\qquad\qquad \forall i:\ u_i \geq 0,\ \forall \omega \neq \omega_0:\ s(\omega) \geq 0,\ \beta \geq 0$ and $\mu \geq 0$.

To find an initial interior feasible point of (P4'), first choose $\lambda_i < 0$ such that $\sum_{i=1}^{n} r_i \lambda_i < 1$. The $u_i > 0$ are then fixed by eq. (P4'c), and μ is fixed by eq. (P4'e). Note that $\mu = 1 - \sum_{i=1}^{n} r_i \lambda_i > 0$ by construction. Substituting $\alpha = -\beta$ into eq. (P4'b), we can then apply theorem 7 to find interior feasible values for β and $s(\omega)$ for all $\omega \neq \omega_0$. Unfortunately we cannot apply lemma 1 to (P4'), because the problem is no longer fully degenerate.

In the next section, we explain how we can generate random sets of desirable gambles that either avoid or do not avoid sure loss. In section 7, we will then benchmark our three methods on those randomly generated sets.

5. Generating sets of desirable gambles via coherent lower previsions

In this section, we first give algorithms for generating coherent previsions (algorithm 1), polyhedral lower previsions (algorithm 2) and linear-vacuous mixtures (algorithm 3), as mentioned earlier. We then discuss how they can be used to generate sets of desirable gambles for benchmarking.

Algorithm 1 Generate a coherent prevision

Input: Set of outcomes Ω
Output: Coherent prevision P on $\mathcal{L}(\Omega)$

Stage 1. Generate a probability mass function p as follows:

      (a) For each ω, sample r_ω uniformly from $(0,1)$.

      (b) For each ω, set $p(\omega) := \dfrac{\ln r_\omega}{\sum_{\omega \in \Omega} \ln r_\omega}$.

Stage 2. Generate a coherent prevision P

      (a) For any $f \in \mathcal{L}(\Omega)$, $P(f) := E_p(f)$ as in eq. (4).

Algorithm 2 Generate a polyhedral lower prevision

Input: Set of outcomes Ω
      k coherent previsions: $Q_1, \ldots, Q_k$ (e.g. obtained by algorithm 1)
Output: Polyhedral lower prevision $\underline{P}$ on $\mathcal{L}(\Omega)$

Stage 1. For any $f \in \mathcal{L}(\Omega)$, $\underline{P}(f) := \min_{j=1}^{k} \{Q_j(f)\}$.

Algorithm 3 Generate a linear-vacuous mixture

Input: Set of outcomes Ω
      $\delta \in (0,1)$ (e.g. sample δ uniformly from $(0,1)$)
      Coherent prevision Q (e.g. generated by algorithm 1)
Output: Linear-vacuous mixture $\underline{P}$

Stage 1. For any $f \in \mathcal{L}(\Omega)$, $\underline{P}(f) := (1-\delta)Q(f) + \delta \inf f$.

We now explain how to generate sets of desirable gambles that avoid sure loss from any given coherent lower previsions. As we will see, if this coherent lower prevision is more generic, then the generated set of desirable gambles will also be more generic. In addition to the analysis presented here, note that we also generated sets of desirable gambles from lower previsions that avoid sure loss but that are not coherent. However, we found no practical difference; see discussion at the end of section 7.

We start by generating a coherent lower prevision $\underline{E}$ on $\mathcal{L}(\Omega)$ e.g. through one of the above algorithms. Next, we generate a finite subset $\mathcal{K}$ of $\mathcal{L}(\Omega)$ and set $\underline{P}(f) := \underline{E}(f)$ for all $f \in \mathcal{K}$. $\underline{P}$ is coherent too because it is the restriction of a coherent lower prevision [11, p. 58]. Therefore, the set $\mathcal{D} := \{f - \underline{P}(f) : f \in \mathcal{K}\}$ avoids sure loss:

Algorithm 4 Generate a set of desirable gambles that avoids sure loss
---
Input: Set of outcomes Ω
      Number of desirable gambles $n := |\mathcal{D}|$
      Coherent lower prevision $\underline{E}$ on $\mathcal{L}(\Omega)$
Output: Finite set of desirable gambles $\mathcal{D}$ that avoids sure loss

Stage 1. Generate $\{f_j \colon j \in \{1, \ldots, n\}\}$:
             for each ω and j, sample $f_j(\omega)$ uniformly from $(0, 1)$.
Stage 2. For each $i \in \{1, \ldots, n\}$, calculate $\underline{E}(f_i)$.
Stage 3. Set $\mathcal{D} \coloneqq \{f_i - \underline{E}(f_i) \colon i \in \{1, \ldots, n\}\}$.

Which type of coherent lower prevision should we use to generate sets of desirable gambles? To answer this question, first we look at $\mathcal{M}_{\underline{E}}$ and its extreme points, for various classes of $\underline{E}$:

(i) Vacuous lower prevision: $\mathcal{M}_{\underline{E}} = \Delta(\Omega)$ and its extreme points are all $0-1$ valued probabilities.
(ii) Coherent previsions: $\mathcal{M}_{\underline{E}} = \text{ext}\, \mathcal{M}_{\underline{E}} = \{p\}$, $p \in \Delta(\Omega)$.
(iii) Polyhedral lower previsions: as in eq. (8), when M is finite, $\mathcal{M}_{\underline{E}}$ is a polyhedron and has a finite set of extreme points.
(iv) Linear-vacuous mixtures: for $p_0 \in \Delta(\Omega)$ and $\delta > 0$, $\mathcal{M}_{\underline{E}} = \{(1 - \delta)p_0 + \delta p, p \in \Delta(\Omega)\}$ and $\text{ext}\, \mathcal{M}_{\underline{E}} = \{(1-\delta)p_0 + \delta p,\ p \text{ is a } 0-1 \text{ valued probability}\}$.
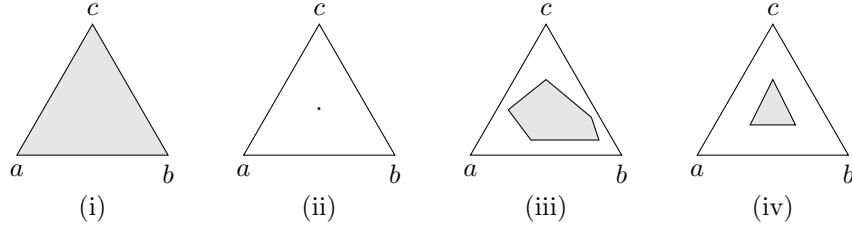
FIGURE 1. Simplex representation of different $\mathcal{M}_{\underline{E}}$ for $\Omega = \{a, b, c\}$: (i) vacuous, (ii) prevision, (iii) polyhedral lower prevision, (iv) linear-vacuous mixture.

Figure 1 shows examples of $\mathcal{M}_{\underline{E}}$ associated with different coherent lower previsions. For polyhedral lower previsions, the number of extreme points is arbitrary (but finite). For linear-vacuous mixtures, the number of extreme points is limited to the number of outcomes, and the shape of its credal set is fixed (up to scale and translation).

When we generate $\mathcal{D} = \{f - \underline{P}(f) \colon f \in \text{dom}\, \underline{P}\}$ as in algorithm 4, the credal set associated with $\mathcal{D}$ is given by

$$(25) \qquad \mathcal{M}_{\mathcal{D}} = \{p \in \Delta(\Omega) \colon \forall f \in \mathcal{D},\ E_p(f) \geq 0\}.$$

$\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\underline{E}}$ are related as follows:

Corollary 1. *Let $\underline{E}$ be a coherent lower prevision on $\mathcal{L}(\Omega)$, let $\underline{P}$ be a restriction of $\underline{E}$ to a finite domain, and let $\mathcal{D} \coloneqq \{f - \underline{P}(f) \colon f \in \text{dom}\, \underline{P}\}$. Then:*

(i) $\mathcal{M}_{\underline{E}} \subseteq \mathcal{M}_{\mathcal{D}}$.
(ii) If $\mathcal{M}_{\underline{E}}$ is a polyhedron, then there is a finite set $\mathcal{K} \subseteq \mathcal{L}(\Omega)$ such that if $\text{dom}\, \underline{P} = \mathcal{K}$, then $\mathcal{M}_{\mathcal{D}} = \mathcal{M}_{\underline{E}}$.
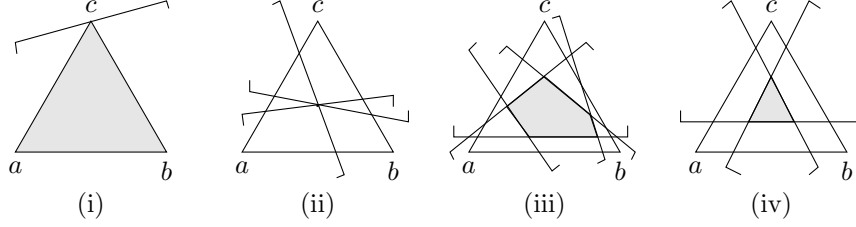
FIGURE 2. Constructing $\mathcal{M}_{\underline{E}}$ by finite half-spaces for $\Omega = \{a, b, c\}$: (i) vacuous, (ii) prevision, (iii) polyhedral lower prevision, (iv) linear-vacuous mixture.

Proof. (i). We find that

$$
(26) \qquad \mathcal{M}_{\underline{E}} = \bigcap_{f \in \mathcal{L}(\Omega)} \{p \in \Delta(\Omega) \colon E_p(f) \geq \underline{E}(f)\}
$$

$$
(27) \qquad = \bigcap_{f \in \mathcal{L}(\Omega)} \{p \in \Delta(\Omega) \colon E_p(f - \underline{E}(f)) \geq 0\}
$$

$$
(28) \qquad \subseteq \bigcap_{f \in \operatorname{dom} \underline{P}} \{p \in \Delta(\Omega) \colon E_p(f - \underline{P}(f)) \geq 0\} = \mathcal{M}_{\mathcal{D}}.
$$

(ii). By (i), we only need to show that $\mathcal{M}_{\mathcal{D}} \subseteq \mathcal{M}_{\underline{E}}$. Since $\mathcal{M}_{\underline{E}}$ is a polyhedron, it is an intersection of a finite number of half-spaces. Therefore there exists an $n \in \mathbb{N}$, vectors $f_1, \ldots, f_n$ and numbers $\alpha_1, \ldots, \alpha_n$ such that

$$
(29) \qquad \mathcal{M}_{\underline{E}} = \bigcap_{i=1}^{n} \{p \in \Delta(\Omega) \colon p \cdot f_i \geq \alpha_i\}
$$

where '$\cdot$' denotes the dot product. Note that

$$
(30) \qquad \underline{E}(f_i) = \min_{p \in \mathcal{M}_{\underline{E}}} p \cdot f_i = \min_{p} \{p \cdot f_i \colon \forall j, \ p \cdot f_j \geq \alpha_j\} \geq \alpha_i.
$$

Set $\mathcal{K} = \{f_1, \ldots, f_n\}$ and $\operatorname{dom} \underline{P} = \mathcal{K}$. Then,

$$
(31) \qquad \mathcal{M}_{\mathcal{D}} = \bigcap_{i=1}^{n} \{p \in \Delta(\Omega) \colon p \cdot f_i \geq \underline{E}(f_i)\}
$$

$$
(32) \qquad \subseteq \bigcap_{i=1}^{n} \{p \in \Delta(\Omega) \colon p \cdot f_i \geq \alpha_i\} = \mathcal{M}_{\underline{E}}.
$$

$\square$

Consequently, if we want $\mathcal{M}_{\mathcal{D}}$ to have a sufficient number of extreme points, we should generate $\mathcal{D}$ using either a polyhedral lower prevision, or at the very least using a linear-vacuous mixture.

Figure 2 visualises the construction of the proof, for various classes of lower previsions. Corollary 1 implies that we do not necessarily gain new extreme points as we add more and more gambles to $\mathcal{D}$. Therefore, algorithm 4 may not be a good way for generating $\mathcal{D}$ if our aim is to keep adding new extreme points with every gamble we add. The algorithm in the next section addresses precisely this issue.

6. Sequentially generating sets of desirable gambles

Consider a set $\mathcal{E} = \{f_1, \ldots, f_n\}$ that avoids sure loss. How can we add another gamble, say f, for which $\mathcal{E} \cup \{f\}$ either still avoids sure loss, or not?

6.1. Generating sets of desirable gambles that do not avoid sure loss.
Given a gamble g, we first find the range of values for α such that $\mathcal{E} \cup \{g - \alpha\}$ avoids sure loss. By the condition of avoiding sure loss in definition 1, we obtain a constraint on the values of α as follows: for all $n \in \mathbb{N}$, all $\lambda_1, \ldots, \lambda_n \geq 0$, and all $f_1, \ldots, f_n \in \mathcal{E}$,

$$(33) \qquad \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) + g(\omega) \right) \geq \alpha.$$

The infimum of this upper bound is precisely $\overline{E}_{\mathcal{E}}(g)$. So, we proved:

Corollary 2. *Let $\mathcal{E}$ avoid sure loss and let $g \in \mathcal{L}(\Omega)$. $\mathcal{E} \cup \{g - \alpha\}$ avoids sure loss if and only if $\alpha \leq \overline{E}_{\mathcal{E}}(g)$.*

Hence, if we set $\alpha > \overline{E}_{\mathcal{E}}(g)$, then $\mathcal{E} \cup \{g - \alpha\}$ does not avoid sure loss:

Algorithm 5 Generate a set of desirable gambles that does not avoid sure loss

---

Input: Set of outcomes Ω
  A set of desirable gambles $\mathcal{E}$ that avoids sure loss
  $\delta > 0$ (e.g. sample δ uniformly from $(0, 1)$)
Output: A set of desirable gambles $\mathcal{D}$ that does not avoid sure loss.

Stage 1. For each $\omega \in \Omega$, sample $g(\omega)$ uniformly from $(0, 1)$.
Stage 2. Solve the following linear program:

$$(34) \qquad \min \quad \beta$$

$$(35) \qquad \text{s. t.} \quad \forall \omega \in \Omega \colon \sum_{f_i \in \mathcal{E}} f_i(\omega)\lambda_i - \beta \leq -g(\omega), \quad \lambda_i \geq 0 \quad (\beta \text{ free}).$$

Stage 3. Set $\mathcal{D} := \mathcal{E} \cup \{g - \beta - \delta\}$.

Note that sets of gambles generated by algorithm 5 only contain a single gamble that violate consistency. Therefore, they are the most computationally challenging sets to detect not avoiding sure loss. Consequently, they are the most suitable sets for benchmarking, as any measurable improvement on these cases implies an at least as large improvement on any simpler cases.

6.2. Generating sets of desirable gambles that avoid sure loss.
Consider a coherent lower prevision $\underline{Q}$. The set $\mathcal{E} := \{f - \underline{Q}(f) \colon f \in \text{dom}\,\underline{Q}\}$ then avoids sure loss. Let $g \in \mathcal{L}(\Omega) \setminus \text{dom}\,\underline{Q}$. By corollary 2, we know that the larger set $\mathcal{D} := \mathcal{E} \cup \{g - \alpha\}$ still avoids sure loss as long as $\alpha \leq \overline{E}_{\mathcal{E}}(g)$. Note that the number of extreme points can decrease after adding $\{g - \alpha\}$, as shown in fig. 3.

How should we choose α to avoid reducing the number of extreme points? If $\underline{P}$ is a coherent extension of $\underline{Q}$ to $\text{dom}\,\underline{Q} \cup \{g\}$, then $\mathcal{M}_{\underline{P}}$ must have at least as many extreme points as $\mathcal{M}_{\underline{Q}}$, because, by coherence, for every $f \in \text{dom}\,\underline{Q}$, $\underline{P}(f)$ and $\underline{Q}(f)$ must be achieved at some extreme point of $\mathcal{M}_{\underline{P}}$ and $\mathcal{M}_{\underline{Q}}$, respectively [14, p. 126]. But because $\underline{P}(f) = \underline{Q}(f)$ for all these gambles f, it cannot be that $\mathcal{M}_{\underline{P}}$ has fewer extreme points than $\mathcal{M}_{\underline{Q}}$, because otherwise $\underline{P}(f) > \underline{Q}(f)$ for at least one gamble f. Hence, the number of extreme points does not decrease if we keep coherence.
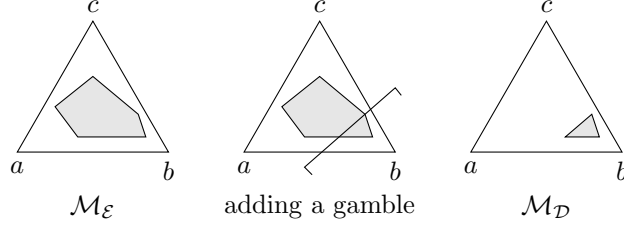
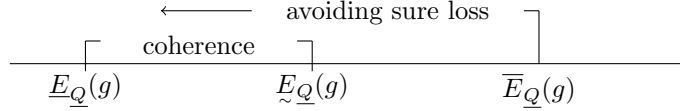FIGURE 3. Simplex representation of a credal sets after adding a gamble.

FIGURE 4. Ranges of avoiding sure loss and coherence

Theorem 8. *(adapted from [14, p. 126]) Let $\underline{Q}$ be a coherent lower prevision and let $g \in \mathcal{L}(\Omega) \setminus \operatorname{dom} \underline{Q}$. Let $\underline{P}$ be an extension of $\underline{Q}$ to $\operatorname{dom} \underline{Q} \cup \{g\}$. Then $\underline{P}$ is coherent if and only if $\underline{P}(g) \in [\underline{E}_Q(g), \underset{\sim}{E}_Q(g)]$, where*

$$(36) \quad \underset{\sim}{E}_Q(g) := \inf_{f_i \in \operatorname{dom} \underline{Q}, \, \lambda_i \geq 0} \left\{ \max_{\omega \in \Omega} \left(g(\omega) + \sum_{i=1}^{n} \lambda_i (f_i(\omega) - \underline{Q}(f_i))) - \lambda_0(f_0(\omega) - \underline{Q}(f_0)) \right) \right\}.$$

Figure 4 shows ranges of avoiding sure loss and coherence of g given a coherent lower prevision $\underline{Q}$.

We can calculate $\underline{E}_Q(g)$ by solving a single linear programming problem. However, $\underset{\sim}{E}_Q(g)$ cannot be obtained by solving just a single linear programming problem: we have to solve a separate linear programming problem for every $f_0 \in \operatorname{dom} \underline{Q}$, where each separate linear program is very similar to and has the same size as the linear program for calculating $\overline{E}_Q(g)$.

So, instead of finding $\underset{\sim}{E}_Q(g)$, we calculate $\underline{E}_Q(g)$ and $\overline{E}_Q(g)$ (or $\overline{E}_\mathcal{E}(g)$ where $\mathcal{E} = \{f - \underline{Q}(f) \colon f \in \operatorname{dom} \underline{Q}\}$) since for each of them, we solve only one linear programming problem. Next, we choose a very small number δ and set $\underline{P}(g) := (1 - \delta)\underline{E}_Q(g) + \delta\overline{E}_Q(g)$. Then, $\underline{P}(g)$ is slightly larger than $\underline{E}_Q(g)$, but $\underline{P}(g)$ is still less than $\overline{E}_Q(g)$. Therefore $\mathcal{E} \cup \{g - \underline{P}(g)\}$ still avoids sure loss as we want. This approach is summarised in algorithm 6. Because only the natural extension of $\underline{Q}$ is used, we only need $\underline{Q}$ to avoid sure loss.

Algorithm 6 Generate a set of desirable gambles that avoids sure loss
---
Input: Set of outcomes Ω
      A lower prevision $\underline{Q}$ that avoids sure loss (e.g. Algorithms 1, 2 or 3)
      A set $\mathcal{E} = \{f - \underline{Q}(f) \colon f \in \operatorname{dom} \underline{Q}\}$ that avoids sure loss
      $\delta \in (0, 1)$
Output: A larger set of desirable gambles $\mathcal{D}$ that avoids sure loss
  Stage 1. For each $\omega \in \Omega$, sample $g(\omega)$ uniformly from $(0, 1)$.

Stage 2. Calculate $\overline{E}_{\underline{Q}}(g)$ by solving

$$(37) \qquad \min \quad \beta$$

$$(38) \qquad \text{subject to} \quad \forall \omega \in \Omega \colon \sum_{i=1}^{n} (f_i(\omega) - \underline{Q}(f_i))\lambda_i - \beta \leq -g(\omega)$$

$$(39) \qquad \lambda_i \geq 0,\ f_i(\omega) - \underline{Q}(f_i) \in \mathcal{E} \qquad (\beta \text{ free})$$

Stage 3. Calculate $\underline{E}_{\underline{Q}}(g)$ by solving

$$(40) \qquad \max \quad \gamma$$

$$(41) \qquad \text{subject to} \quad \forall \omega \in \Omega \colon \sum_{i=1}^{n} (f_i(\omega) - \underline{Q}(f_i))\lambda_i + \gamma \leq g(\omega)$$

$$(42) \qquad \lambda_i \geq 0,\ f_i(\omega) - \underline{Q}(f_i) \in \mathcal{E} \qquad (\gamma \text{ free}).$$

Stage 4. Set $\underline{P}(g) := (1 - \delta)\underline{E}_{\underline{Q}}(g) + \delta\overline{E}_{\underline{Q}}(g)$.
Stage 5. Set $\mathcal{D} := \mathcal{E} \cup \{g - \underline{P}(g)\}$.

Note that we will not use algorithm 6 in our benchmarking. Instead, we use a combination of algorithm 2 and algorithm 4, which generates constraints via a set of probability mass functions, as this is computationally faster. We state algorithm 6 for the sake of completeness, as an alternative algorithm for generating constraints directly.

7. Numerical results

To benchmark our theoretical result, in this section, we generate two types of random sets of desirable gambles: sets that avoid sure loss, and sets that do not. For each type, we consider $|\mathcal{D}| = 2^i$ for $i \in \{1, 2, \ldots, 8\}$ and $|\Omega| = 2^j$ for $j \in \{1, 2, \ldots, 8\}$. Random sets that avoid sure loss are generated as follows:

(1) We use algorithm 1 to generate k coherent previsions. We fixed $k = 2^5$, as we observed that varying k has little impact on the results.
(2) From these k coherent previsions, we use algorithm 2 to generate a polyhedral lower prevision.
(3) We use algorithm 4, with this polyhedral lower prevision, to generate a random set that avoids sure loss.

Next, starting from a set $\mathcal{E}$ that avoids sure loss, we generate a set that does not avoid sure loss using algorithm 5 with $\delta = 0.05$.

For each random set of desirable gambles, we use the algorithms for checking avoiding sure loss discussed earlier in the paper. Table 1 gives an overview of these different algorithms. Note that the primal-dual method simultaneously solves the primal and the dual problems.

To compare these three methods, we wrote our own implementation of the affine scaling and the primal-dual methods. We used an implementation of the revised simplex method written by **(author?)** [10]. Indeed, the revised simplex method is mathematically equivalent to the standard simplex method, but is much more efficient and numerically stable as it applies sparse matrix manipulations [4, §3.7]. We used the revised simplex method to solve both problems (P3) and (D3). For the affine scaling method, a standard version is used for solving (D4'), while an improved version, which includes the extra stopping criterion and our mechanism

Linear programs	Methods		
	Simplex	Affine scaling	Primal-dual
(P3)	✓	✓	✓
(D3)	✓		
(D4')		✓	✓

TABLE 1. List of different methods for checking avoiding sure loss. Note that the primal-dual method also solves the dual problem simultaneously, e.g. (D5) for (P3) and (P4') for (D4').

for calculating feasible starting points, is used to solve (P3). For the primal-dual method, an improved version that has the extra stopping criterion and the mechanism for calculating feasible starting points is used for solving (P3) and (D5), and another improved version that has only the mechanism for calculating feasible starting points is used to solve (D4') and (P4').

For each method, we run the algorithm twice to remove any warm-up effects that can happen in the first run, and we only measure the corresponding computational time taken in the second run. We repeat the process 1000 times and present a summary of the results in figs. 5 and 6.

Figures 5 and 6 show the average computational time taken during each method when checking avoiding sure loss. In the left column, the sets of desirable gambles avoid sure loss while in the right column, they do not avoid sure loss. In fig. 5 each row represents a different number of desirable gambles, and the horizontal axis represents the number of outcomes. In fig. 6, each row represents a different number of outcomes, and the horizontal axis represents the number of desirable gambles. In both figures, the vertical axis shows the computational time. The computational time is averaged over 1000 random sets of desirable gambles. The error bars on the figures represent approximate 95% confidence intervals on the mean computation time. These are barely visible because of the large sample size, except in some rare cases where we observed large variability in the simplex method (possibly due to the numerical issues that we discussed earlier in the paper).

In the avoiding sure loss case, note that the sets of desirable gambles are always generated from coherent lower previsions. However, in some applied problems, this may not be the case. Therefore, we ran one further experiment where we introduced a negative bias, which removes coherence whilst still avoiding sure loss. Specifically, in Stage 3 of algorithm 4, we set $\mathcal{D} \coloneqq \{f_i - \underline{E}(f_i) + \eta(f_i) \colon i \in \{1, \ldots, n\}\}$, for some $\eta(f_i) > 0$. We considered two scenarios: (i) we uniformly sampled each $\eta(f_i)$ from the open $(0, 1)$ interval, and (ii) we fixed each $\eta(f_i) \coloneqq 0.01$. This made no practical difference. In particular, the plots in figs. 5 and 6 for the avoiding sure loss case remained nearly identical, with no change in general conclusions.

8. DISCUSSION AND CONCLUSION

In this study, we discussed and improved the simplex, the affine scaling and the primal-dual methods to efficiently solve linear programming problems for checking whether sets of desirable gambles avoid sure loss or not. To measure our improvements, we also gave several algorithms for generating random sets of desirable gambles that either avoid or do not avoid sure loss.
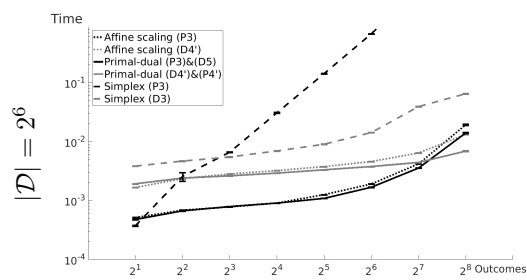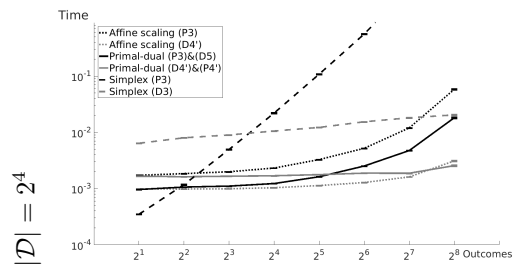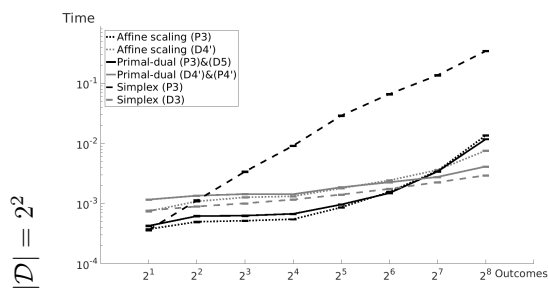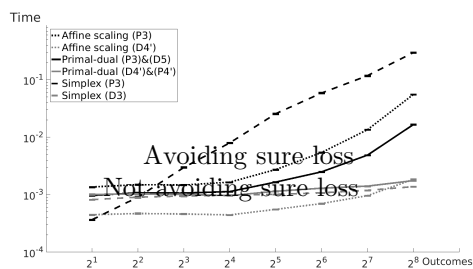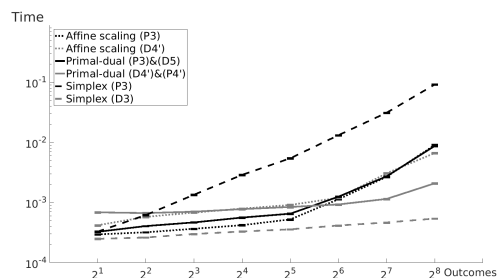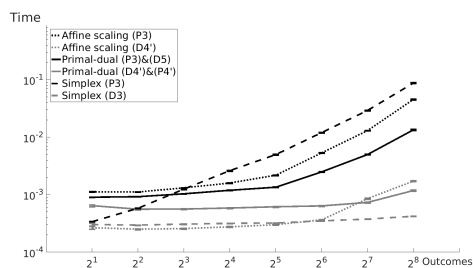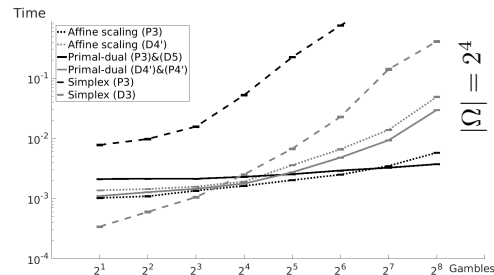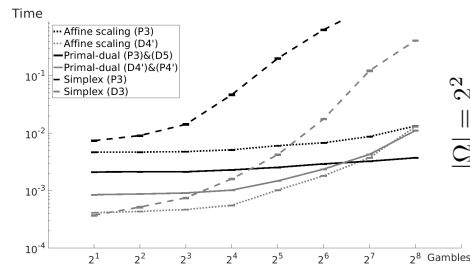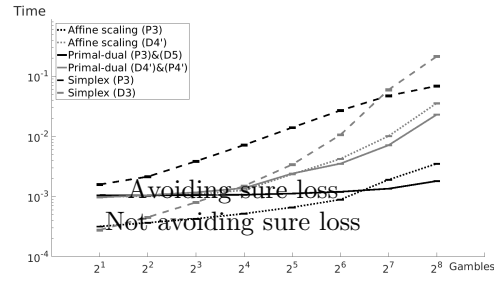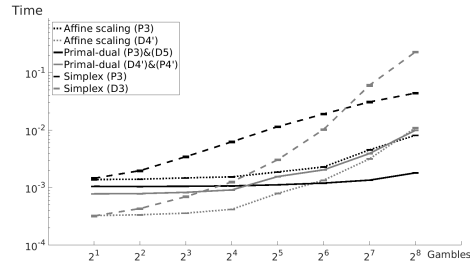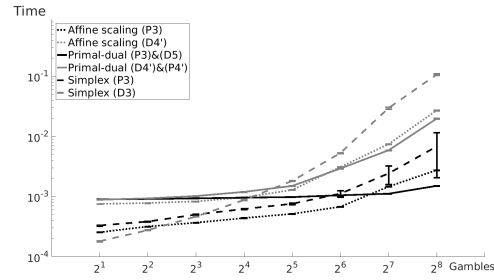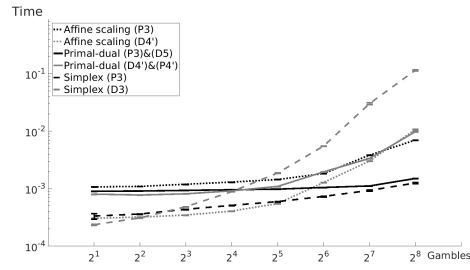
Building further from [8], we studied linear programming problems and two improvements of these methods, namely, (i) an extra stopping criterion, and (ii) a simple and quick algorithm for finding feasible starting points in these methods. We compared the impact of these improvements on the three methods for solving linear programming problems.

These improvements benefit all applied problems as they reduce the computational burden of the original algorithms. Our benchmarking study quantified these benefits for a wide range of situations. In case of not avoiding sure loss, we tested the hardest case where only a single gamble violates consistency. Any positive computational gain in these cases implies an at least as large gain for more general applied cases where multiple gambles violate consistency.

According to our numerical results, the relative performance of the three methods depends on the number of desirable gambles and the number of outcomes. Specifically, if the number of outcomes is much larger than the number of desirable gambles, then solving either (D3) or (D4') is faster than solving (P3). However, if the number of outcomes is much less than the number of desirable gambles, then we prefer to solve (P3). When the two numbers are of similar magnitude, there is no clear difference.

In the results, the primal-dual and the affine scaling methods outperform the simplex method in most cases. The simplex method can find a basic feasible starting point easily, but it cannot apply the extra stopping criterion. Therefore, the simplex method is not a good choice for checking avoiding sure loss.

On the other hand, the affine scaling and the primal-dual methods can benefit from these improvements. Specifically, when we solve (P3), these two methods can apply the extra stopping criterion and a simple mechanism to calculate feasible starting points. In this case, the primal-dual method performs very well, especially when we do not avoid sure loss and the number of desirable gambles is large.

When we solve (D4'), these two interior-point methods can easily find feasible starting points. In this case, the affine scaling method performs very well in small problems whilst the primal-dual method does better when the problems are bigger.

Overall, if problems are small, then there is no big difference in the time taken to solve either (P3) or (D4'), and there is also no big difference between the performance of the methods. When the problems are large, the primal-dual method is the best choice. In this case, if the number of desirable gambles is large, then we solve (P3), and if the number of outcomes is large, then we solve (D4').

In future work, we will test our improved methods with some real applications and explore an algorithm for choosing ω_0 in theorem 6 for large problems.

Appendix A. Proofs

Proof of theorem 6. We only show that $\mathcal{D}$ avoids sure loss if and only if the optimal value of (P2) is zero since the proof that the dual problem, (D2), has feasible solutions follows immediately by the strong duality theorem [4, p.59].

Firstly, by lemma 1, the optimal value of (P2) is either zero or unbounded. Next, we show that if $\mathcal{D}$ avoids sure loss, then the optimal value of (P2) is zero, and vice versa. Note that eq. (P2b) can be written as

$$(43) \qquad \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) \right) \leq \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha.$$

Suppose $\mathcal{D}$ avoids sure loss, then by definition 2, for all n, all $\lambda_1, \ldots, \lambda_n \geq 0$, and $f_1, \ldots, f_n \in \mathcal{D}$,

$$(44) \qquad 0 \leq \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) \right).$$

So, by eq. (43),

$$(45) \qquad 0 \leq \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha.$$

So, the optimal value is non-negative. Now, by putting $\lambda_i = 0$ for all i, and $\alpha = 0$, we obtain

$$(46) \qquad \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha = 0.$$

Therefore, the optimal value of (P2) is zero.

Conversely, suppose $\mathcal{D}$ does not avoid sure loss. There are non-negative $\lambda_1, \ldots, \lambda_n$ such that

$$(47) \qquad \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) \right) < 0.$$

Set

$$(48) \qquad s = \sup_{\omega \in \Omega} \left(\sum_{i=1}^{n} \lambda_i f_i(\omega) \right)$$

and choose

$$(49) \qquad \alpha = s - \sum_{i=1}^{n} \lambda_i f_i(\omega_0).$$

Now we have $\alpha \geq 0$ and

$$(50) \qquad \forall \omega \neq \omega_0 \colon \sum_{i=1}^{n} \lambda_i f_i(\omega) \leq \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha = s < 0.$$

This means that

$$(51) \qquad \sum_{i=1}^{n} \lambda_i f_i(\omega_0) + \alpha < 0$$

is a feasible value of (P2). By lemma 1, the optimal value is unbounded.   □

References

[1] Kurt M. Anstreicher. Linear programming: Karmarkar projective algorithm. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization, Second Edition*, pages 1889–1891. Springer, New York, USA, 2009.

[2] Thomas Augustin, Frank P. A. Coolen, Gert De Cooman, and Matthias C. M. Troffaes, editors. *Introduction to Imprecise Probabilities*. Wiley Series in Probability and Statistics. Wiley, 2014. URL: `http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470973811.html`.

[3] W. H. Cunningham. Theoretical properties of the network simplex method. *Mathematics of Operations Research*, 4(2):196–208, 1979. `doi:10.1287/moor.4.2.196`.

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

[5] C.J. Goh and X.Q.Yang. *Duality in optimization and variational inequalities*. Taylor and Francis, London, 2002.

[6] Frederick S. Hillier and Gerald J. Lieberman. *Inroduction to operations research*. McGraw-Hill, 2001.

[7] Katta G. Murty. Complexity of degeneracy. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization, Second Edition*, pages 419–425. Springer, New York, USA, 2009.

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

[9] Erik Quaeghebeur. *A Propositional CONEstrip Algorithm*, pages 466–475. Springer International Publishing, 2014. `doi:10.1007/978-3-319-08852-5_48`.

[10] Gilbert Strang. Introduction to linear algebra. *MATLAB Central File Exchange*, 1.0, 2002, Update 30 Aug 2002. URL: `https://uk.mathworks.com/matlabcentral/fileexchange/2166-introduction-to-linear-algebra?focused=5039479&tab=function`.

[11] Matthias C. M. Troffaes and Gert de Cooman. *Lower Previsions*. Wiley Series in Probability and Statistics. Wiley, 2014. URL: `http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470723777.html`.

[12] T. Tsuchiya and M. Muramatsu. Global convergence of a long-step affine scaling algorithm for degenerate linear programming problems. *Research Memorandum 423, The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106, Japan*, January 1992; revised September, 1992.

[13] Robert J. Vanderbei. *Linear Programming: Foundations and Extensions, Second edition*. Springer, 2001.

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

[15] Peter Walley, Renato Pelessoni, and Paolo Vicig. Direct algorithms for checking consistency and making inferences from conditional probability assessments. *Journal of Statistical Planning and Inference*, 126:119–151, 2004.

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

[17] 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`.

Durham University, Department of Mathematical Sciences, UK
E-mail address: `nawapon.nakharutai@durham.ac.uk`

Durham University, Department of Mathematical Sciences, UK
E-mail address: `matthias.troffaes@durham.ac.uk`

Durham University, Department of Mathematical Sciences, UK
E-mail address: `c.c.d.s.caiado@durham.ac.uk`