CONEstrip

Wieger Wesselink

October 4, 2024

1 Introduction

This document describes the implementation of the CONEstrip and the Propositional CONEstrip algorithm, see [2] and [3].

2 Notations

Let P be an arbitrary set, and $A = \{a_1, \ldots, a_n\}$ a finite set. Then we use the notation P^A as a shorthand for the Cartesian product $P^{|A|}$. Furthermore we use the notation $\lambda \in P^A$ as a shorthand notation for $\lambda = [\lambda_{a_1}, \ldots, \lambda_{a_n}]$.

Definition 1: For a set V the *indicator function* $\mathbb{1}_V$ is defined as

$$\mathbb{1}_V(x) = \begin{cases} 1 & \text{if } x \in V \\ 0 & \text{otherwise.} \end{cases}$$

For $v \in V$ we define $\mathbb{1}_v = \mathbb{1}_{\{v\}}$.

3 Cones

Definition 2: A *field* $(F, +, \cdot)$ is a set F together with two binary operations on F called addition and multiplication. These operations are required to satisfy the field axioms:

- Associativity of addition and multiplication: $a + (b + c) = (a + b) + c$, and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.
- Commutativity of addition and multiplication: $a + b = b + a$, and $a \cdot b = b \cdot a$.
- Additive and multiplicative identity: there exist two different elements 0 and 1 in F such that $a + 0 = a$ and $a \cdot 1 = a$.
- Additive inverses: for every a in F, there exists an element in F, denoted $-a$, called the additive inverse of a, such that $a + (-a) = 0$.
- Multiplicative inverses: for every $a \neq 0$ in F, there exists an element in F, denoted by a^{-1} or $1/a$, called the multiplicative inverse of a, such that $a \cdot a^{-1} = 1$.
- Distributivity of multiplication over addition: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Definition 3: A field $(F, +, \cdot)$ together with a (strict) total order $<$ on F is an *ordered field* if the order satisfies the following properties for all $a, b, c \in F$:

1. if $a < b$ then $a + c < b + c$

2. if $0 < a$ and $0 < b$ then $0 < a \cdot b$.

Definition 4: A *vector space* $(V, +, \cdot)$ over a field F is a set V together with two operations, addition and scalar multiplication, that satisfy the eight axioms listed below.

- Associativity of vector addition: $u + (v + w) = (u + v) + w$.
- Commutativity of vector addition: $u + v = v + u$.
- Identity element of vector addition. There exists an element $0 \in V$, called the zero vector, such that $v + 0 = v$ for all $v \in V$.
- Inverse elements of vector addition. For every $v \in V$, there exists an element $-v \in V$, called the additive inverse of v, such that $v + (-v) = 0$.
- Compatibility of scalar multiplication with field multiplication: $a(bv) = (ab)v$.
- Identity element of scalar multiplication: $1v = v$, where 1 denotes the multiplicative identity in F.
- Distributivity of scalar multiplication with respect to vector addition: $a(u + v) = au + av$.
- Distributivity of scalar multiplication with respect to field addition: $(a + b)v = av + bv$.

where u, v and w denote arbitrary vectors in V, and a and b denote scalars in F.

Definition 5: A *cone* is a subset C of a vector space V over an ordered field F. A cone C is a convex cone if $\alpha x + \beta y$ belongs to C, for any positive scalars α, β, and any x, y in C. A cone C is convex if and only if $C + C \subseteq C$.

Definition 6: A subset $U \subseteq \mathbb{A}^d$ is *open* (in the norm topology) if either U is empty or for every point $a \in U$, there is some (small) open ball $B(a, \epsilon)$ contained in U. A subset $C \subseteq \mathbb{A}^d$ is *closed* iff $\mathbb{A}^d - C$ is open.

4 Gambles

Let $\Omega = \{\omega_1, \ldots, \omega_n\}$ be a non-empty, finite set of outcomes (possibility space). N.B. In this document we restrict ourselves to finite sets of outcomes. Many of the definitions in this document are taken from [1].

Definition 7: A *gamble on* Ω is a bounded mapping from Ω to $\mathbb{R}$, i.e., $f : \Omega \to \mathbb{R}$. Gambles are used to represent an agent's beliefs and information.

Let $\Omega = \{a, b, c, d\}$. An example of a gamble is: $f(a) = 3$, $f(b) = -2$, $f(c) = 5$, $f(d) = 10$. If an agent accepts a gamble f, then the value $f(\omega)$ represents the reward she would obtain if ω is the true unknown value (this value can be negative and then it represents a loss).

Let $\mathcal{G} \subseteq \mathbb{R}^n$ denote the set of all gambles defined on Ω. For $f, g \in \mathcal{G}$, let $f \geq g$ mean that $f(\omega) \geq g(\omega)$ for all $\omega \in \Omega$, and let $f > g$ mean that $f \geq g$ and $f(\omega) > g(\omega)$ for some $\omega \in \Omega$.

Definition 8: A subset $\mathcal{D}$ of $\mathcal{G}$ is said to be a *coherent set of desirable gambles* relative to $\mathcal{G}$ when it satisfies the following four axioms (N.B. slightly adapted from [1]):

D1. $0 \in \mathcal{D}$,

D2. If $f \geq 0$, then $f \in \mathcal{D}$ (**Accept partial gain**),

D3. If $f \in \mathcal{D}$ and $\lambda \geq 0$, then $\lambda f \in \mathcal{D}$ (**Positive Scale Invariance**),

D4. If $f \in \mathcal{D}$ and $g \in \mathcal{D}$, then $f + g \in \mathcal{D}$ (**Combination**).

Definition 9: For bounded gambles f, g a set of bounded gambles $\mathcal{D}$ is *coherent* when the following four conditions are satisfied (see [4]):

A1. If $f < 0$, then $f \notin \mathcal{D}$ (**Avoid partial loss**),

A2. If $f \geq 0$, then $f \in \mathcal{D}$ (**Accept partial gain**)

A3. If $f \in \mathcal{D}$ and $\lambda \geq 0$, then $\lambda f \in \mathcal{D}$ (**Positive Scale Invariance**)

A4. If $f \in \mathcal{D}$ and $g \in \mathcal{D}$, then $f + g \in \mathcal{D}$ (**Combination**)

Axioms A1 and A2 are rationality conditions: positive payments are desirable (A2) while negative payments are not (A1). Axiom A3 says that the desirability of a gamble is unchanged by the introduction of a positive scale and axiom A4 says that desirability is additive.

Definition 10: A *coherent set* $\mathcal{D}$ *of almost desirable gambles* is a set of gambles which satisfies the following axioms (the first one is a modification of the corresponding axiom for desirable gambles. The new version is called avoiding sure loss), see [1]:

D1'. $-1 \notin D$ (**Avoid sure loss** (?)),

D2. If $f \geq 0$, then $f \in \mathcal{D}$ (**Accept partial gain**),

D3. If $f \in \mathcal{D}$ and $\lambda \geq 0$, then $\lambda f \in \mathcal{D}$ (**Positive Scale Invariance**),

D4. If $f \in \mathcal{D}$ and $g \in \mathcal{D}$, then $f + g \in \mathcal{D}$ (**Combination**).

D5. if $f + \epsilon \in \mathcal{D}, \forall \epsilon > 0$, then $f \in \mathcal{D}$.

Almost desirable gambles avoid uniform loss, but not partial loss.

Definition 11: The *lower prevision induced by* $\mathcal{D}$ is the function $\underline{P} : \mathcal{G} \to \mathbb{R}$ defined as follows:

$$\underline{P}(f) = \sup\{\mu \in \mathbb{R} \mid f - \mu \in \mathcal{D}\}.$$

The *upper prevision induced by* $\mathcal{D}$ is the function $\overline{P} : \mathcal{G} \to \mathbb{R}$ defined as follows:

$$\overline{P}(f) = \inf\{\mu \in \mathbb{R} \mid \mu - f \in \mathcal{D}\}.$$

The lower prevision for a gamble f is the supremum acceptable buying price for f, meaning that I am inclined to buy it for $\underline{P}(f) - \epsilon$ for any $\epsilon > 0$.

Definition 12: A *credal set* is a closed and convex set of probability measures.

A set of desirable gambles $\mathcal{D}$ defines a credal set: $P_{\mathcal{D}} = \{P \mid P[X] \geq 0, \forall X \in \mathcal{D}\}$.

Definition 13: If $\mathcal{G}$ is an arbitrary set of gambles, then the set of all gambles obtained by applying axioms D2, D3, and D4 is called the *set of gambles generated by* $\mathcal{G}$ and it is denoted by $\overline{\mathcal{G}}$. If this set is coherent ($0 \notin \overline{\mathcal{G}}$) then it will be called its *natural extension* (the minimum coherent set containing $\mathcal{G}$). If $0 \in \mathcal{G}$ we will say that $\mathcal{G}$ is incoherent (**N.B. This contradicts with definition 9!**). If $f < 0$ and $f \in \overline{\mathcal{G}}$ we will say that $\mathcal{G}$ does not avoid partial loss.

Definition 14: Let $\mathcal{R}$ be a finite set of finite subsets of $\mathcal{G}$. Then we define the *general cone* $\underline{\mathcal{R}}$ as the smallest set that contains $\mathcal{R}$ and that satisfies axioms D2, D3 and D4.

5 Feasibility problems

Let $\Omega = \{\omega_1, \ldots, \omega_n\}$. Let $\mathcal{A} = \{g_1, \ldots, g_m\}$ with $g_i \in \Omega \to \mathbb{R}^n$, $(1 \leq i \leq m)$ be a finite set of almost desirable gambles.

5.1 Avoiding sure loss

The feasibility problem below checks if $\mathcal{A}$ incurs sure loss, see [2]:

$$\begin{aligned} &\text{find} && \lambda \in \mathbb{R}^{\mathcal{A}} \\ &\text{subject to} && \sum_{g \in \mathcal{A}} \lambda_g \cdot g \leq -1 \text{ and } \lambda \geq 0. \end{aligned} \tag{1}$$

By introducing slack variables μ, this can be rewritten into the equivalent problem

$$\begin{aligned} &\text{find} && \lambda \in \mathbb{R}^{\mathcal{A}} \text{ and } \mu \in \mathbb{R}^{\Omega} \\ &\text{subject to} && \sum_{g \in \mathcal{A}} \lambda_g \cdot g + \sum_{\omega \in \Omega} \mu_\omega \cdot 1_\omega = 0 \text{ and } \lambda \geq 0 \text{ and } \mu \geq 1. \end{aligned} \tag{2}$$

5.2 Avoiding sure loss of a lower prevision

Let $\underline{P} : \mathcal{K} \to \mathbb{R}$ be a lower prevision with $\mathcal{K}$ finite, then checking whether $\underline{P}$ incurs sure loss amounts to solving (1) for $\mathcal{A} = \{h - \underline{P}(h) \mid h \in \mathcal{K}\}$.

5.3 Calculating the lower prevision

The lower prevision for a gamble $f \in \mathcal{G}$ is calculated using the linear program below (natural extension), see [2]:

$$\begin{aligned} &\text{maximize} && \alpha \in \mathbb{R} \\ &\text{subject to} && f - \alpha \geq \sum_{g \in \mathcal{A}} \lambda_g \cdot g \text{ and } \lambda \geq 0. \end{aligned} \tag{3}$$

By introducing slack variables μ, this can be rewritten into the equivalent problem

$$\begin{aligned} &\text{maximize} && \alpha \in \mathbb{R} \\ &\text{subject to} && \sum_{i=1}^{m} \lambda_i \cdot g_i + \sum_{j=1}^{n} \mu_j \cdot 1_j + \alpha = f \text{ and } \lambda \geq 0 \text{ and } \mu \geq 0. \end{aligned} \tag{4}$$

5.4 Determining interior points of a cone

To determine if $f \in \mathcal{A}$ is an interior point in the cone generated by $\mathcal{A}$, we define the following optimization problem:

$$\begin{aligned} &\text{find} && \lambda \in \mathbb{R}^{\mathcal{A} \setminus \{f\}} \\ &\text{subject to} && f = \sum_{g \in \mathcal{A} \setminus \{f\}} \lambda_g \cdot g \text{ and } \lambda > 0. \end{aligned} \tag{5}$$

Definition 15: We call f an *interior point* of the cone generated by $\mathcal{A}$ iff equation 5 has a solution.

5.5 Coin examples

Example 16: Consider an experiment with a coin, with $\Omega = \{H, T\}$ and probabilities $p(H) = p(T) = 0.5$. We represent the indicator function $\mathbb{1}_H$ by the tuple $[\mathbb{1}_H(H), \mathbb{1}_H(T)] = [1, 0]$ and the indicator function $\mathbb{1}_T$ by the tuple $[\mathbb{1}_T(H), \mathbb{1}_T(T)] = [0, 1]$. Let $\mathcal{R} = \{\{\mathbb{1}_H\}, \{\mathbb{1}_T\}\}$ be the generators of an open cone.

To compute the lower prevision $\underline{\mathbb{1}_H}$ take $f = \mathbb{1}_H - \alpha$ with $\alpha \in \mathbb{R}$, then

$$\mathbb{E}[f] = p(H)f(H) + p(T)f(T) = 0.5(1-\alpha) + 0.5(-\alpha) = 0.5 - \alpha.$$

The minimum is reached for $\alpha = 0.5$, hence $\underline{\mathbb{1}_H} = 0.5$.

To compute the lower prevision $\underline{\mathbb{1}_T}$ take $f = \mathbb{1}_T - \alpha$ with $\alpha \in \mathbb{R}$, then

$$\mathbb{E}[f] = p(H)f(H) + p(T)f(T) = 0.5(-\alpha) + 0.5(1-\alpha) = 0.5 - \alpha.$$

Again the minimum is reached for $\alpha = 0.5$, hence $\underline{\mathbb{1}_T} = 0.5$.

Example 17: Consider an experiment with a coin, with $\Omega = \{H, T\}$ and suppose we know the following about the probabilities:

$$\begin{cases} p(H) > \frac{1}{3} \\ p(T) > \frac{1}{5} \end{cases}$$

From this we derive

$$\begin{cases} \frac{1}{3} < p(H) \leq \frac{4}{5} \\ \frac{1}{5} < p(T) \leq \frac{2}{3}. \end{cases}$$

To compute the lower prevision of a gamble f we solve $\mathbb{E}[f] > 0$, or equivalently $\mathbb{E}[f - \alpha] = 0$, with $\alpha \geq 0$.

For $f = \mathbb{1}_H$ we calculate

$$\mathbb{E}[f - \alpha] = (1-\alpha) \cdot p(H) - \alpha \cdot p(T) > (1-\alpha) \cdot \frac{1}{3} - \alpha \cdot \frac{2}{3} = \frac{1}{3} - \alpha.$$

Thus we get $\underline{\mathbb{1}_H} = \frac{1}{3}$.

For $f = \mathbb{1}_T$ we calculate

$$\mathbb{E}[f - \alpha] = (-\alpha) \cdot p(H) + (1-\alpha) \cdot p(T) > (-\alpha) \cdot \frac{4}{5} + (1-\alpha) \cdot \frac{1}{5} = \frac{1}{3} - \alpha.$$

Thus we get $\underline{\mathbb{1}_T} = \frac{1}{5}$.

6 The CONEstrip algorithm

Let Ω be a possibility space, and let $\mathcal{G} \subseteq \Omega \to \mathbb{R}^n$ be the set of all gambles.

Definition 18 (Cone generator)**:** A *cone generator* is a finite set of gambles.

Definition 19 (General cone)**:** A *general cone* is a finite set of cone generators.

6.1 The optimization problems of the CONEstrip algorithm

The CONEstrip algorithm is an algorithm that determines whether a gamble belongs to a given general cone. It depends on some optimization problems that we will define in this section. The first instance of the optimization problems is solve-conestrip1. In three iterations it is rewritten into solve-conestrip4, which is suitable to be used in the CONEstrip algorithm.

Let $\mathcal{R}$ be a general cone, and let $f \in \mathcal{G}$ be a gamble. Let Ω_Γ and Ω_Δ be sets of events such that $\Omega_\Gamma \cup \Omega_\Delta = \Omega$.

Definition 20: We define solve-conestrip1$(\mathcal{R}, f, \Omega_\Gamma, \Omega_\Delta)$ as an arbitrary solution (λ, ν) of the following optimization problem, or $(\perp, \perp)$ if no solution exists. See also [3], formula (1).

$$\begin{aligned} &\text{maximize} \quad \sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \\ &\text{such that} \quad \begin{cases} \lambda_{\mathcal{D}} \in [0,1] \text{ and } \nu_{\mathcal{D}} \in (\mathbb{R}_{>0})^{\mathcal{D}} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \\ \sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} = 1 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g(\omega)) \leq f(\omega) \quad \text{for all } \omega \in \Omega_\Gamma \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g(\omega)) \geq f(\omega) \quad \text{for all } \omega \in \Omega_\Delta, \end{cases} \end{aligned} \tag{6}$$

In many practical cases we have $\Omega_\Gamma = \Omega_\Delta = \Omega$. Then the equations simplify to

$$\begin{cases} \sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} = 1 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g) = f \end{cases}$$

Remark 21: Without loss of generality we can replace the constraint $\sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} = 1$ with $\sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \geq 1$.

Definition 22: We define solve-conestrip2$(\mathcal{R}, f, \Omega_\Gamma, \Omega_\Delta)$ as as an arbitrary solution (λ, τ, σ) of the following optimization problem, or $(\perp, \perp, \perp)$ if no solution exists. See also [3], formula (2).

$$\begin{aligned} &\text{maximize} \quad \sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \\ &\text{such that} \quad \begin{cases} \lambda_{\mathcal{D}} \in [0,1] \text{ and } \tau_{\mathcal{D}} \in (\mathbb{R}_{\geq 1})^{\mathcal{D}} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \text{ and } \sigma \in \mathbb{R}_{\geq 1} \\ \sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \geq 1 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \tau_{\mathcal{D},g} \cdot g(\omega)) \leq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Gamma \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \tau_{\mathcal{D},g} \cdot g(\omega)) \geq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Delta, \end{cases} \end{aligned} \tag{7}$$

where $\tau_{\mathcal{D}} = \sigma \nu_{\mathcal{D}}$.

Definition 23: We define $\mathsf{solve\text{-}conestrip3}(\mathcal{R}, f, \Omega_\Gamma, \Omega_\Delta)$ as as an arbitrary solution (λ, μ, σ) of the following optimization problem, or $(\perp, \perp, \perp)$ if no solution exists. See also [3], formula (3).

$$\begin{aligned} &\text{maximize} \quad \sum_{\mathcal{D}\in\mathcal{R}} \lambda_\mathcal{D} \\ &\text{such that} \quad \begin{cases} \lambda_\mathcal{D} \in [0,1] \text{ and } \mu_\mathcal{D} \in (\mathbb{R}_{\geq 0})^\mathcal{D} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \text{ and } \sigma \in \mathbb{R}_{\geq 1} \\ \sum_{\mathcal{D}\in\mathcal{R}} \lambda_\mathcal{D} \geq 1 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g(\omega)) \leq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Gamma \\ \sum_{\mathcal{D}\in\mathcal{R}} (\sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g(\omega)) \geq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Delta, \\ \lambda_\mathcal{D} \leq \mu_{\mathcal{D},g} \leq \lambda_\mathcal{D}\mu_{\mathcal{D},g} \quad \text{for all } \mathcal{D} \in \mathcal{R}, g \in \mathcal{D}. \end{cases} \end{aligned} \tag{8}$$

Notice that now $\lambda_\mathcal{D} \in \{0,1\}$ for any solution, functioning as a switch between $\mu_\mathcal{D} = 0, \tau_\mathcal{D} = 0$ and $\mu_\mathcal{D} = 1, \tau_\mathcal{D} \in (\mathbb{R}_{\geq 1})^\mathcal{D}$, so that $\mu_\mathcal{D}/\sigma$ effectively behaves as $\lambda_\mathcal{D}\nu_\mathcal{D}$.

Definition 24: We define $\mathsf{solve\text{-}conestrip4}(\mathcal{R}, f, \Omega_\Gamma, \Omega_\Delta)$ as as an arbitrary solution (λ, μ, σ) of the following optimization problem, or $(\perp, \perp, \perp)$ if no solution exists. See also [3], formula (4).

$$\begin{aligned} &\text{maximize} \quad \sum_{\mathcal{D}\in\mathcal{R}} \lambda_\mathcal{D} \\ &\text{such that} \quad \begin{cases} \lambda_\mathcal{D} \in [0,1] \text{ and } \mu_\mathcal{D} \in (\mathbb{R}_{\geq 0})^\mathcal{D} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \text{ and } \sigma \in \mathbb{R}_{\geq 1} \\ \sum_{\mathcal{D}\in\mathcal{R}} \lambda_\mathcal{D} \geq 1 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g(\omega)) \leq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Gamma \\ \sum_{\mathcal{D}\in\mathcal{R}} (\sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g(\omega)) \geq \sigma f(\omega) \quad \text{for all } \omega \in \Omega_\Delta \\ \lambda_\mathcal{D} \leq \mu_{\mathcal{D},g} \quad \text{for all } \mathcal{D} \in \mathcal{R}, g \in \mathcal{D}. \end{cases} \end{aligned} \tag{9}$$

Algorithm 1 Determining membership of a general cone

Input: A general cone $\mathcal{R}$, a gamble $f \in \mathcal{G}$ and two sets $\Omega_\Gamma, \Omega_\Delta$ with $\Omega_\Gamma \cup \Omega_\Delta = \Omega$.
Output: Whether or not f belongs to the general cone $\mathcal{R}$.

```
function CONESTRIP(R, f, Ω_Γ, Ω_Δ)
    while true do
        (λ, μ, σ) := solve-conestrip4(R, f, Ω_Γ, Ω_Δ)
        if (λ, μ, σ) = (⊥, ⊥, ⊥) then
            return false
        if ∀_{D∈R} : λ_D = 0 ⇒ (∀_{g∈D} : μ_{D,g} = 0) then
            return true
        R := {D ∈ R | λ_D ≠ 0}
```

6.2 Generating test cases

Let $\mathcal{R}$ be a set of finite subsets of $\mathcal{G}$. To generate test cases for the CONEstrip algorithm, we will iteratively do the following.

1. Randomly choose a finite subset $\mathcal{A} \in \mathcal{R}$ of at least dimension two.
2. Generate a lower dimensional cone $\mathcal{A}'$ that is contained in the border of the cone generated by $\mathcal{A}$.
3. Add $\mathcal{A}'$ to $\mathcal{R}$.

For step 2, we do the following:

1. Let $\mathcal{B}$ be obtained from $\mathcal{A}$ by removing all interior points from $\mathcal{A}$.
2. Randomly choose $|\mathcal{B}| - 1$ elements from the set $\{\sum_{g \in \mathcal{B}} \lambda^{\mathcal{B}} \cdot g \mid \lambda^{\mathcal{B}} \geq 0 \wedge \exists g \in \mathcal{B} : \lambda_g^{\mathcal{B}} = 0\}$. They generate a lower dimensional cone that is contained in the border of $\mathcal{A}$.

7 The Propositional Context

In practice the possibility space Ω can be very large. To deal with this, we will now define a symbolic representation of Ω by means of propositional formulas. To this end we assume that the possibility space Ω is a subset of $\{0,1\}^m$ for some $m > 0$.

Definition 25 (Propositional sentences)**:** Let B be an ordered set of boolean variables. The set P_B of *propositional sentences over* B is inductively defined using

1. if $b \in B$ then $b \in P_B$
2. if $\phi \in P_B$ then $\neg\phi \in P_B$
3. if $\phi_1, \phi_2 \in P_B$ then $\phi_1 \vee \phi_2 \in P_B$
4. if $\phi_1, \phi_2 \in P_B$ then $\phi_1 \wedge \phi_2 \in P_B$.

Definition 26 (Propositional sentence function)**:** A propositional sentence ψ over boolean variables $B = \{b_1, \ldots, b_m\}$ can be interpreted as a function $\psi : \{0,1\}^m \to \{0,1\}$ using

$$\psi(\beta) = \psi[b_1 := \beta_1, \ldots, b_m := \beta_m] \tag{10}$$

for $\beta \in \{0,1\}^m$.

Example 27: Let $\mathcal{G} = \mathbb{R}^4$, let $\Omega = \{00, 01, 10, 11\}$ and let $B = \{b_1, b_2\}$ be a set of two boolean variables. Now take $\phi \in \mathcal{G}$ be the gamble that is defined as

$$\begin{aligned}
\phi(00) &= 1\\
\phi(01) &= 0\\
\phi(10) &= 1\\
\phi(11) &= 1.
\end{aligned}$$

Then the propositional sentence $\hat{\phi} = b_1 \vee \neg b_2$ defines exactly the same function as ϕ using the interpretation in definition 26.

7.1 Basic functions

We assume that a finite set of so called basic functions $\Phi \subset \mathcal{G}$ is given that forms a basis of $\mathcal{G}$. By this we mean that any gamble $g \in \mathcal{G}$ can be written as a linear combination of these basic functions (a.k.a. indicator functions):

$$g = \sum_{\phi \in \Phi} g_\phi \phi \tag{11}$$

with $g_\phi \in \mathbb{R}$ for $\phi \in \Phi$. We restrict the basic functions to be boolean valued, i.e.

$$\phi \in \Omega \to \{0,1\} \quad \text{for } \phi \in \Phi.$$

This makes it possible to define the basic functions $\phi \in \Phi$ as propositional sentences over a set of boolean variables $B = \{b_1, \ldots, b_m\}$, as is demonstrated in example 27. The possibility space Ω and the sets Ω_Γ and Ω_Δ can also be defined by means of propositional sentences ψ, ψ_Γ and ψ_Δ over B using

$$\begin{aligned}
\Omega &= \{\beta \in \{0,1\}^m \mid \psi(\beta) = 1\} &(12)\\
\Omega_\Gamma &= \{\beta \in \Omega \mid \psi_\Gamma(\beta) = 1\} &(13)\\
\Omega_\Delta &= \{\beta \in \Omega \mid \psi_\Delta(\beta) = 1\} &(14)
\end{aligned}$$

7.2 Propositional gambles

Let Φ be a set of basic functions of the gambles $\mathcal{G}$.

Definition 28 (Propositional gamble)**:** Let $g \in \mathcal{G}$ be a gamble. The corresponding *propositional gamble* over Φ is the vector of coordinates $(g_{\phi_1}, \ldots, g_{\phi_k}) \in \mathbb{R}^k$ with respect to the basis Φ, as defined in equation (11).

Definition 29 (Propositional cone generator)**:** A *propositional cone generator* over Φ is a finite set of propositional gambles over Φ.

Definition 30 (Propositional general cone)**:** A *propositional general cone* over Φ is a finite set of propositional cone generators over Φ.

Definition 31 (Propositional basis)**:** A *propositional basis* of the set of gambles $\mathcal{G}$ is a finite set Φ of propositional sentences such that the gambles associated with Φ according to the interpretation in definition (26) form a basis of $\mathcal{G}$.

7.3 The subproblems of the propositional CONEstrip algorithm

The propositional CONEstrip algorithm depends on three subproblems that we will define in this section. Let $B = \{b_1, \ldots, b_m\}$ and $C = \{c_1, \ldots, c_k\}$ be sets of boolean variables, let ψ be a propositional sentence over B and let $\Phi = \{\phi_1, \ldots, \phi_k\}$ be a propositional basis over B.

Definition 32: We define $\textsf{solve-propositional-conestrip1}(\psi, B, C, \Phi)$ as an arbitrary solution (β, γ) of the following satisfiability problem, or $(\bot, \bot)$ if no solution exists:

$$\text{find } (\beta, \gamma) \qquad \text{such that} \quad \psi(\beta) \wedge (\forall_{1 \leq i \leq k} : \phi_i(\beta) \leftrightarrow \gamma_i) \tag{15}$$

Definition 33: Let Γ and Δ be subsets of $\{0,1\}^k$. We define $\textsf{solve-propositional-conestrip2}(\mathcal{R}, f, \Gamma, \Delta, \Phi)$ as an arbitrary solution $(\lambda, \mu, \sigma, \kappa)$ of the following optimization problem, or $(\bot, \bot, \bot, \bot)$ if no solution exists.

$$\begin{aligned}
&\text{maximize} && \sum_{\mathcal{D} \in \mathcal{R}} \lambda_{\mathcal{D}} \\
&\text{for} && \lambda_{\mathcal{D}} \in [0,1] \text{ and } \mu_{\mathcal{D}} \in (\mathbb{R}_{\geq 0})^{\mathcal{D}} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \text{ and } \sigma \in \mathbb{R}_{\geq 1} \text{ and } \kappa_\phi \in \mathbb{R} \text{ for all } \phi \in \Phi \\
&\text{such that} && \begin{cases}
\sum_{\mathcal{D} \in \mathcal{R}} \lambda_{\mathcal{D}} \geq 1 \\
\lambda_{\mathcal{D}} \leq \mu_{\mathcal{D},g} \quad \text{for all } \mathcal{D} \in \mathcal{R}, g \in \mathcal{D} \\
\kappa_\phi = \sum_{\mathcal{D} \in \mathcal{R}} \big(\sum_{g \in \mathcal{D}} \mu_{\mathcal{D},g} \cdot g_\phi \big) - \sigma f_\phi \quad \text{for all } \phi \in \Phi \\
\sum_{i=1}^{k} \kappa_i \gamma_i \leq 0 \quad \text{for all } \gamma \in \Gamma \\
\sum_{i=1}^{k} \kappa_i \delta_i \geq 0 \quad \text{for all } \delta \in \Delta.
\end{cases}
\end{aligned} \tag{16}$$

Definition 34: Let $\kappa \in \mathbb{R}^k$. We define $\textsf{solve-propositional-conestrip3-max}(\psi, \kappa, B, C, \Phi)$ as an arbitrary solution (β, γ) of the optimization problem

$$\text{maximize } \sum_{i=1}^{k} \kappa_i \gamma_i \qquad \text{such that} \quad \psi(\beta) \wedge (\forall_{1 \leq i \leq k} : \phi_i(\beta) \leftrightarrow \gamma_i) \tag{17}$$

and $\textsf{solve-propositional-conestrip3-min}(\psi, \kappa, B, C, \Phi)$ as an arbitrary solution (β, γ) of the linear programming problem

$$\text{minimize } \sum_{i=1}^{k} \kappa_i \gamma_i \qquad \text{such that} \quad \psi(\beta) \wedge (\forall_{1 \leq i \leq k} : \phi_i(\beta) \leftrightarrow \gamma_i) \tag{18}$$

Algorithm 2 Determining membership of a general cone

Input: A propositional general cone $\mathcal{R}$, a propositional gamble f, a finite set of boolean variables B, a propositional basis $\Phi = \{\phi_1, \ldots, \phi_k\}$ over B, and propositional sentences ψ, ψ_Γ and ψ_Δ over B such that $\psi_\Gamma \wedge \psi_\Delta \to \psi$.

Output: Whether or not f belongs to the general cone $\mathcal{R}$.

function $\textsc{PropositionalCONEstrip}(\mathcal{R}, f, B, \Phi, \psi, \psi_\Gamma, \psi_\Delta)$
  $C := \{c_1, \ldots, c_k\}$ ▷ C is a set of fresh boolean variables
  $\Gamma := \emptyset$
  $(\beta, \gamma) := \textsf{solve-propositional-conestrip1}(\psi \wedge \psi_\Gamma, B, C, \Phi)$
  if $\gamma \neq \bot$ **then**
    $\Gamma := \{\gamma\}$
  $\Delta := \emptyset$
  $(\beta, \delta) := \textsf{solve-propositional-conestrip1}(\psi \wedge \psi_\Delta, B, C, \Phi)$
  if $\delta \neq \bot$ **then**
    $\Delta := \{\delta\}$
  while true **do**
    $(\lambda, \mu, \sigma, \kappa) := \textsf{solve-propositional-conestrip2}(\mathcal{R}, f, \Gamma, \Delta, \Phi)$
    if $(\lambda, \mu, \sigma, \kappa) = (\bot, \bot, \bot, \bot)$ **then**
      return false
    $\mathcal{R} := \{\mathcal{D} \in \mathcal{R} \mid \lambda_\mathcal{D} \neq 0\}$
    $\gamma := 0^k$
    $\delta := 0^k$
    if $\Gamma \neq \emptyset$ **then**
      $(\beta, \gamma) := \textsf{solve-propositional-conestrip3-max}(\psi \wedge \psi_\Gamma, \kappa, B, C, \Phi)$
      $\Gamma := \Gamma \cup \{\gamma\}$
    if $\Delta \neq \emptyset$ **then**
      $(\beta, \delta) := \textsf{solve-propositional-conestrip3-min}(\psi \wedge \psi_\Delta, \kappa, B, C, \Phi)$
      $\Delta := \Delta \cup \{\delta\}$
    if $\sum_{i=1}^{k} \kappa_i \gamma_i \leq 0 \leq \sum_{i=1}^{k} \kappa_i \delta_i$ and $\forall_{\mathcal{D} \in \mathcal{R}} : \lambda_\mathcal{D} = 0 \Rightarrow (\mu_{\mathcal{D},g} = 0 \text{ for all } g \in \mathcal{D})$ **then**
      return true

---

8 Optimization problems

Let $\mathcal{R}$ be a general cone and f a gamble. Let $\Omega = \{\omega_1, \ldots, \omega_N\}$ be the set of elementary events. Without loss of generality we will assume that $\Omega = \{1, \ldots, N\}$. We define

$$1_\omega(x) = \begin{cases} 1 & \text{if } x = \omega \\ 0 & \text{otherwise} \end{cases}$$

and

$$1_\Omega(x) = \begin{cases} 1 & \text{if } x \in \Omega \\ 0 & \text{otherwise.} \end{cases}$$

In our implementation we represent a gamble f by the vector $(f(\omega_1), \ldots, f(\omega_N))$. Hence 1_{ω_i} is represented by the i-th unit vector $e_i \in \mathbb{R}^N$, and 1_Ω is represented by 1_N, the vector containing N ones. The zero gamble 0 is represented by 0_N, the vector containing N zeroes.

For vectors $x, y \in \mathbb{R}^n$ we define

$$\begin{aligned} x \gtrdot y &\equiv x_i > y_i \quad (1 \le i \le n) \\ x \geq y &\equiv x_i \geq y_i \quad (1 \le i \le n) \\ x > y &\equiv x \geq y \wedge x \neq y \end{aligned}$$

The operators $\geq$ and $\gtrdot$ can be easily generalized to higher dimensional variables.

Definition 35: We define $\mathsf{optimize\text{-}find}(\mathcal{R}, f, \{b^i, c^i\}_{i\in I}, \Omega)$ as a solution μ of problem (20) or $\perp$ if no solution exists. The default formulation of $\mathsf{optimize\text{-}find}$ is in terms of variables λ and ν.

$$\begin{aligned} &\text{find} && (\lambda, \nu) \text{ with } \lambda \in \mathbb{R}^{\mathcal{R}} \text{ and } \nu_{\mathcal{D}} \in \mathbb{R}^{\mathcal{D}} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \\ &\text{such that} && \begin{cases} \lambda > 0 \text{ and } \nu \gtrdot 0 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g) = f \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot b^i_{\mathcal{D},g}) = c^i \quad \text{for all } i \in I \end{cases} \end{aligned} \tag{19}$$

The variables λ and ν can be combined using $\mu_{\mathcal{D},g} = \lambda_{\mathcal{D}} \nu_{\mathcal{D},g}$, which gives us the following system:

$$\begin{aligned} &\text{find} && \mu \text{ with } \mu_{\mathcal{D}} \in \mathbb{R}^{\mathcal{D}} \text{ for all } \mathcal{D} \text{ in } \mathcal{R} \\ &\text{such that} && \begin{cases} \mu \geq 0 \text{ and } \exists \mathcal{D} \in \mathcal{R} : \mu_{\mathcal{D}} \gtrdot 0 \\ \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = f \\ \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot b^i_{\mathcal{D},g} = c^i \quad \text{for all } i \in I \end{cases} \end{aligned} \tag{20}$$

Definition 36: We define $\mathsf{optimize\text{-}maximize}(\mathcal{R}, f, a, \{b^i, c^i\}_{i\in I}, \Omega)$ as a solution μ of problem (22) or $\perp$ if no solution exists. The default formulation of $\mathsf{optimize\text{-}maximize}$ is in terms of variables λ and ν, with $\lambda \in \mathbb{R}^{\mathcal{R}}$ and $\nu_{\mathcal{D}} \in \mathbb{R}^{\mathcal{D}}$ for all $\mathcal{D}$ in $\mathcal{R}$.

$$\begin{aligned} &\text{maximize} && \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot a_{\mathcal{D},g}) \\ &\text{such that} && \begin{cases} \lambda > 0 \text{ and } \nu \gtrdot 0 \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g) = f \\ \sum_{\mathcal{D}\in\mathcal{R}} (\lambda_{\mathcal{D}} \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot b^i_{\mathcal{D},g}) = c^i \quad \text{for all } i \in I \end{cases} \end{aligned} \tag{21}$$

The variables λ and ν can be combined using $\mu_{\mathcal{D},g} = \lambda_{\mathcal{D}}\nu_{\mathcal{D},g}$, which gives us the following system:

$$\begin{array}{ll} \text{maximize} & \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot a_{\mathcal{D},g} \\ \text{such that} & \begin{cases} \mu \geq 0 \text{ and } \exists\mathcal{D} \in \mathcal{R} : \mu_{\mathcal{D}} > 0 \\ \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = f \\ \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot b^i_{\mathcal{D},g} = c^i \quad \text{for all } i \in I \end{cases} \end{array} \tag{22}$$

We define

$$\mathsf{optimize\text{-}maximize\text{-}value}(\mathcal{R}, f, a, \{b^i, c^i\}_{i\in I}, \Omega) = \begin{cases} \infty & \text{if } \mu = \bot \\ \sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot a_{\mathcal{D},g} & \text{otherwise,} \end{cases}$$

where $\mu = \mathsf{optimize\text{-}maximize}(\mathcal{R}, f, a, \{b^i, c^i\}_{i\in I}, \Omega)$. In other words, optimize-maximize-value returns the value of the goal function for a solution μ.

8.1 Lower prevision functions

In our use case lower prevision functions are simply real-valued functions defined on a set of gambles.

Definition 37: Let $\underline{P}$ be a lower prevision function defined on the finite set of gambles $\mathcal{K} \subseteq \mathcal{G}$. Then we define

$$\mathsf{lower\text{-}prevision\text{-}assessment}(\underline{P}) = \{h - \underline{P}(h) \mid h \in \mathcal{K}\}$$

Definition 38: Let $\underline{P}$ be a conditional lower prevision function defined on the finite set $\mathcal{N} \subseteq \mathcal{G} \times \Omega^*$. Then we define

$$\mathsf{conditional\text{-}lower\text{-}prevision\text{-}assessment}(\underline{P}, \Omega) = \{((h - \underline{P}(h|B)) \odot 1_B, B) \mid (h, B) \in \mathcal{N}\}$$

Example 39 (Lower prevision functions)**:** Let p be a mass function on Ω i.e.

$$p(\omega) \geq 0 \text{ for } \omega \in \Omega \quad \text{and} \quad \sum_{\omega\in\Omega} p(\omega) = 1,$$

Let m be a mass function on 2^Ω (i.e. the set of subsets of Ω), with the constraint that $m(\emptyset) = 0$. Let ε be a given positive value. The following functions are practical examples of lower prevision functions defined on a set of gambles $\mathcal{K} \subseteq \mathcal{G}$.

$$\mathsf{linear\text{-}lower\text{-}prevision\text{-}function}(p, \mathcal{K}) = \underline{P}, \quad \text{with } \underline{P}(f) = \sum_{\omega\in\Omega} p(\omega) f(\omega) \tag{23}$$

$$\mathsf{linear\text{-}vacuous\text{-}lower\text{-}prevision\text{-}function}(p, \delta, \mathcal{K}) = \underline{P}, \quad \text{with } \underline{P}(f) = (1-\delta) \sum_{\omega\in\Omega} p(\omega) f(\omega) + \delta \min_{\omega\in\Omega} f(\omega), \tag{24}$$

$$\mathsf{belief\text{-}lower\text{-}prevision\text{-}function}(m, \mathcal{K}) = \underline{P}, \quad \text{with } \underline{P}(f) = \sum_{S\subseteq\Omega} m(S) \min_{\omega\in S} f(\omega) \tag{25}$$

8.2 Optimization problems

Definition 40 (Incurring sure loss)**:** Let $\mathcal{R}$ be a general cone. Then we define

$$\mathsf{incurs\text{-}sure\text{-}loss\text{-}cone}(\mathcal{R}, \Omega) \equiv \mathsf{optimize\text{-}find}(\mathcal{R}, 0, \emptyset, \Omega) \neq (\bot, \bot)$$

Let $\underline{P}$ be a lower prevision function with $\mathcal{A} = \textsf{lower-prevision-assessment}(\underline{P})$. Then we define

$$\textsf{incurs-sure-loss}(\underline{P}, \Omega) \equiv \textsf{incurs-sure-loss-cone}(\mathcal{R}, \Omega),$$

with

$$\mathcal{R} = \textsf{sure-loss-cone}(\mathcal{A}, \Omega)$$

and

$$\textsf{sure-loss-cone}(\mathcal{A}, \Omega) = \{\{1_\omega \mid \omega \in \Omega\} \ \cup \ \mathcal{A} \setminus \{0\}\}.$$

Definition 41 (Unconditional natural extension)**:** Let $\underline{P}$ be a lower prevision function, let f be a gamble and let $\mathcal{A} = \textsf{lower-prevision-assessment}(\underline{P})$. We define

$$\textsf{natural-extension}(\mathcal{A}, f, \Omega) \equiv \textsf{optimize-maximize-value}(\mathcal{R}, f, a, \emptyset, \Omega),$$

where

$$\begin{cases} \mathcal{R} & = \textsf{natural-extension-cone}(\mathcal{A}, \Omega) \\ a & = \textsf{natural-extension-objective}(\mathcal{R}, \Omega) \end{cases}$$

with

$$\textsf{natural-extension-cone}(\mathcal{A}, \Omega) = \{\{g\} \mid g \in \mathcal{A}\} \ \cup \ \{\{1_\Omega\}, \{-1_\Omega\}, \{0\}\} \ \cup \ \{\{1_\omega\} \mid \omega \in \Omega\}$$

and $\textsf{natural-extension-objective}(\mathcal{R}, \Omega)$ is the function a defined for all $\mathcal{D} \in \mathcal{R}$ and $g \in \mathcal{D}$ as

$$a_{\mathcal{D},g} = \begin{cases} 1 & \text{if } \mathcal{D}, g = \{1_\Omega\}, 1_\Omega \\ -1 & \text{if } \mathcal{D}, g = \{-1_\Omega\}, -1_\Omega \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

Definition 42 (Coherence)**:** Let $\underline{P}$ be a lower prevision function defined on the finite set of gambles $\mathcal{K} \subseteq \mathcal{G}$ with $\mathcal{A} = \textsf{lower-prevision-assessment}(\underline{P})$. We define

$$\textsf{is-coherent}(\underline{P}, \Omega) \equiv \forall f \in \mathcal{K} : \underline{P}(f) = \textsf{natural-extension}(\mathcal{A}, f, \Omega)$$

Definition 43 (Incurring partial loss)**:** Let $\underline{P}$ be a conditional lower prevision function defined on the finite set $\mathcal{N} \subseteq \mathcal{G} \times \Omega^*$ with

$$\mathcal{B} = \textsf{conditional-lower-prevision-assessment}(\underline{P}, \Omega).$$

Then we define

$$\textsf{incurs-partial-loss}(\underline{P}, \Omega) = \textsf{optimize-find}(\mathcal{R}, 0, \emptyset, \Omega) \neq (\bot, \bot)$$

where

$$\mathcal{R} = \textsf{partial-loss-cone}(\mathcal{B}, \Omega)$$

with

$$\textsf{partial-loss-cone}(\mathcal{B}, \Omega) = \{\{g, 1_B\} \mid (g, B) \in \mathcal{N} \wedge g \neq 0\} \ \cup \ \{\{1_\omega\} \mid \omega \in \Omega\}.$$

Definition 44 (Conditional natural extension)**:** Let $\underline{P}$ be a conditional lower prevision function defined on the finite set $\mathcal{N} \subseteq \mathcal{G} \times \Omega^*$, let f be a gamble, let C be an event, and let $\mathcal{B} = \textsf{conditional-lower-prevision-assessment}(\underline{P}, \Omega)$. Then we define

$$\textsf{conditional-natural-extension}(\mathcal{B}, f, C, \Omega) \equiv \textsf{optimize-maximize-value}(\mathcal{R}, f \odot 1_C, a, \emptyset, \Omega),$$

where

$$\begin{cases} \mathcal{R} & = \textsf{conditional-natural-extension-cone}(\mathcal{B}, C, \Omega) \\ a & = \textsf{natural-extension-objective}(\mathcal{R}, \Omega) \end{cases}$$

with

$$\textsf{conditional-natural-extension-cone}(\mathcal{B}, C, \Omega) = \{\{g, 1_B\} \mid (g, B) \in \mathcal{N}\} \cup \{\{1_C\}, \{-1_C\}, \{0\}\} \cup \{\{1_\omega \mid \omega \in \Omega\}\}.$$

8.3 Test case 1

Let Ω be a set of elementary events and let $\mathcal{K} \subseteq \mathcal{G}$ be an arbitrary, finite set of gambles. Let δ be a given error magnitude.

This test consists of the following steps:

1. Randomly generate a mass function p on Ω.
2. Let $\underline{P}_p = \textsf{linear-vacuous-lower-prevision-function}(p, \delta, \mathcal{K})$.
3. Calculate $b := \textsf{incurs-sure-loss}(\underline{P}_\varepsilon, \Omega)$.
4. Check $\neg b$.

Note that if $\delta = 0$, we replace step 2 by $\underline{P}_p = \textsf{linear-lower-prevision-function}(p, \mathcal{K})$.

Definition 45 (Perturbations)**:** Let $0 < \varepsilon \leq 1$ be an error magnitude and $\mathcal{K}$ a finite set of gambles. Then we define a class of randomly generated perturbations in $\mathcal{K} \to \mathbb{R}$ as follows:

$$\textsf{generate-lower-prevision-perturbation}(\mathcal{K}, \varepsilon) = \mathcal{Q}, \quad \text{with } \mathcal{Q}(f) = \pm\, \varepsilon\, (\max_{\omega\in\Omega} f(\omega) - \min_{\omega\in\Omega} f(\omega)) \cdot \delta, \tag{27}$$

where $\delta \sim U(0, 1)$.

Definition 46 (Clamped sum)**:** When applying a perturbation to a lower prevision function, it can make sense to limit the perturbation by taking a clamped sum. Let P and Q be two lower prevision functions, then we define

$$\textsf{lower-prevision-clamped-sum}(P, Q) = R, \quad \text{with } R(f) = \textsf{clamp}(P(f) + Q(f), \min(f), \max(f)), \tag{28}$$

where

$$\textsf{clamp}(x, \mathit{min\text{-}value}, \mathit{max\text{-}value}) = \max(\min(x, \mathit{max\text{-}value}), \mathit{min\text{-}value}).$$

8.4 Test case 2

Let Ω be a set of elementary events and let $\mathcal{K} \subseteq \mathcal{G}$ be an arbitrary, finite set of gambles. Let δ be a given error magnitude. Let $E = [\varepsilon_1, \ldots, \varepsilon_q]$ be a range of small positive values.

This test consists of the following steps:

1. Randomly generate a mass function p on Ω.
2. Let $\underline{P}_p = \textsf{linear-vacuous-lower-prevision-function}(p, \delta, \mathcal{K})$.
3. For $\varepsilon \in E$ do
   (a) Generate a perturbation $\mathcal{Q}_\varepsilon := \textsf{generate-lower-prevision-perturbation}(\mathcal{K}, \varepsilon)$.
   (b) Let $\underline{P}_\varepsilon = \textsf{lower-prevision-clamped-sum}(\underline{P}_p, \mathcal{Q}_\varepsilon)$.
   (c) Calculate $\textsf{incurs-sure-loss}(\underline{P}_\varepsilon, \Omega)$.

Note that if $\delta = 0$, we replace step 2 by $\underline{P}_p = \textsf{linear-lower-prevision-function}(p, \mathcal{K})$.

The result of this test is checked manually. For small values of ε we expect that $\textsf{incurs-sure-loss}(\underline{P}_\varepsilon, \Omega)$ has the value false.

8.5 Test case 3

Let Ω be a set of elementary events, and let $\mathcal{K} \subseteq \mathcal{G}$ be an arbitrary, finite set of gambles. This test case is about generating 4-dimensional data sets that can be studied later on. The inputs of a test case are

$$\begin{cases} M & \text{the number of probability mass functions} \\ I & \text{the number of imprecision values (default: 10)} \\ E & \text{the number of error magnitudes (default: 10)} \\ N & \text{the number of repetitions} \end{cases}$$

These four dimensions are identified as "pmf", "imprecision", "errmag" and "repetitions". The output is a set of values

$$\{Q_{m,i,e,n} \mid 1 \leq m \leq M \wedge 1 \leq i \leq I \wedge 1 \leq e \leq E \wedge 1 \leq n \leq N\},$$

where each value $Q_{m,i,e,n}$ is a tuple of two boolean values representing sure loss and coherence.

This test case consists of the following steps:

1. Randomly generate mass functions $p_1, \ldots, p_M$ on Ω.
2. Choose imprecision values $\delta_1, \ldots, \delta_I$ in the interval [0, 1].
3. Choose error magnitudes $\varepsilon_1, \ldots, \varepsilon_E$ in $\mathbb{R}^+$.
4. For each m, i, e, n generate a lower prevision function

   $$\underline{P}_{m,i,e,n} = \mathsf{lower\text{-}prevision\text{-}clamped\text{-}sum}(P, Q),$$

   where

   $$\begin{aligned} P &= \mathsf{linear\text{-}vacuous\text{-}lower\text{-}prevision\text{-}function}(p_m, \delta_i, \mathcal{K}) \\ Q &= \mathsf{generate\text{-}lower\text{-}prevision\text{-}perturbation}(\mathcal{K}, \varepsilon_e) \end{aligned}$$

5. Set $Q_{m,i,e,n} = (\mathsf{incurs\text{-}sure\text{-}loss}(\underline{P}_{m,i,e,n}, \Omega), \mathsf{is\text{-}coherent}(\underline{P}_{m,i,e,n}, \Omega))$

N.B. this test case only generates a dataset. The results are checked manually.

8.6 Test case 4

Let Ω be a finite set of elementary events, and let $\mathcal{K} \subseteq \mathcal{G}$ be a finite set of gambles.

This test consists of the following steps:

1. Randomly generate a mass function p on Ω.
2. Let $\underline{P}_{\mathsf{lower}} = \mathsf{linear\text{-}lower\text{-}prevision\text{-}function}(p, \mathcal{K})$.
3. For $\varepsilon \in \{0, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}\}$ do
   (a) Generate a perturbation $\mathcal{Q}_\varepsilon := \mathsf{generate\text{-}lower\text{-}prevision\text{-}perturbation}(\mathcal{K}, \varepsilon)$.
   (b) Let $\underline{P} = \mathsf{lower\text{-}prevision\text{-}clamped\text{-}sum}(\underline{P}_{\mathsf{lower}}, \mathcal{Q}_\varepsilon)$.
   (c) Calculate $b := \mathsf{incurs\text{-}sure\text{-}loss}(\underline{P}, \Omega)$.
   (d) Calculate $c := \mathsf{is\text{-}coherent}(\underline{P}, \Omega)$.
   (e) Check that $\neg b \vee \neg c$.

References

[1] Inés Couso and Serafín Moral. Sets of desirable gambles and credal sets. *ISIPTA 2009 - Proceedings of the 6th International Symposium on Imprecise Probability: Theories and Applications*, pages 99–108, 01 2009.

[2] Erik Quaeghebeur. The conestrip algorithm. In Rudolf Kruse, Michael R. Berthold, Christian Moewes, María Ángeles Gil, Przemysław Grzegorzewski, and Olgierd Hryniewicz, editors, *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, pages 45–54, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[3] Erik Quaeghebeur. A propositional conestrip algorithm. In Anne Laurent, Oliver Strauss, Bernadette Bouchon-Meunier, and Ronald R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 466–475, Cham, 2014. Springer International Publishing.

[4] Gregory Wheeler. Discounting desirable gambles. In Andres Cano, Jasper De Bock, Enrique Miranda, and Serafín Moral, editors, *Proceedings of the Twelveth International Symposium on Imprecise Probability: Theories and Applications*, volume 147 of *Proceedings of Machine Learning Research*, pages 331–341–331–341. PMLR, 06–09 Jul 2021.