The CONEstrip Algorithm

Erik Quaeghebeur[1]

Abstract Uncertainty models such as sets of desirable gambles and (conditional) lower previsions can be represented as convex cones. Checking the consistency of and drawing inferences from such models requires solving feasibility and optimization problems. We consider finitely generated such models. For closed cones, we can use linear programming; for conditional lower prevision-based cones, there is an efficient algorithm using an iteration of linear programs. We present an efficient algorithm for general cones that also uses an iteration of linear programs.

Key words: convex cones, consistency, feasibility, inference, linear programming

1 Introduction

Mathematically speaking, frameworks for modeling uncertainty consist of rules that specify what constitutes a within the framework valid model and rules to perform computations with such models. For a number of frameworks under the imprecise probability umbrella [8, 9], checking validity—i.e., the consistency criteria of avoiding sure & partial loss and coherence—and calculating an inference—i.e., natural extension—involves solving feasibility and optimization problems.

We illustrate in Section 2 that the feasibility aspect of these problems essentially boils down to checking whether some vector lies in a *general convex cone*, called *general cone* from now on, a cone that may be closed, open, or *ajar*, i.e., neither open nor closed. For models specified in a finitary way, algorithms to do this for closed and specific general cases can be found in the literature. In Section 4 we present an efficient algorithm for all general finitary instances. But to do this, we first need to make a small detour with Section 3 to discuss how we can represent finitary general cones—and therefore the feasibility and optimization problems that interest us—in a way that is conducive to algorithm formulation.

[1]SYSTeMS Research Group, Ghent University, Belgium Erik.Quaeghebeur@UGent.be

Concepts & Notation. We assume that the *possibility space* Ω is non-empty and finite. *Gambles* are real-valued functions on Ω, i.e., elements of $\mathcal{L} \coloneqq [\Omega \to \mathbb{R}]$. The *indicator* 1_B of an event $B \subseteq \Omega$ is 1 on B and 0 elsewhere; $1_\omega \coloneqq 1_{\{\omega\}}$ for $\omega \in \Omega$. The finite subset relation is $\Subset$. A set superscripted with '$*$' denotes the set of all its finite subsets. Vector inequalities: $>$ ($\geq$) is pointwise (non-)strict; $>$ means "$\geq$ but not =".

2 Problems Solved in the Literature

In this section, we present a number of problems from and solved in the literature. On the one hand, they are meant to make a link with the literature, and on the other hand, we use them to illustrate that these problems essentially boil down to checking whether some vector lies in a cone.

Lower Previsions and Sets of Almost-Desirable Gambles. The most basic consistency criterion for lower previsions and sets of almost desirable gambles is *avoiding sure loss* [8, §2.4 & §3.7.1]. Checking whether a lower prevision $\underline{P} \in [\mathcal{K} \to \mathbb{R}]$, with $\mathcal{K} \Subset \mathcal{L}$, or set of almost desirable gambles $\mathcal{A} \Subset \mathcal{L}$ *incurs sure loss* amounts to solving the feasibility problem below, where in the former case $\mathcal{A} \coloneqq \{h - \underline{P}h : h \in \mathcal{K}\}$:

$$\begin{aligned} \text{find} \quad & \lambda \in \mathbb{R}^{\mathcal{A}}, \\ \text{subject to} \quad & \textstyle\sum_{g \in \mathcal{A}} \lambda_g \cdot g < 0 \quad \text{and} \quad \lambda \geq 0. \end{aligned}$$

We can get an equivalent feasibility problem that can however be solved using linear programming [10, §2.4] by replacing the constraints by

$$\textstyle\sum_{g \in \mathcal{A}} \lambda_g \cdot g \leq -1 \quad \text{and} \quad \lambda \geq 0.$$

By introducing (slack) variables $\mu \in \mathbb{R}^{\Omega}$, these can also be written as

$$\textstyle\sum_{g \in \mathcal{A}} \lambda_g \cdot g + \sum_{\omega \in \Omega} \mu_\omega \cdot 1_\omega = 0 \quad \text{and} \quad \lambda \geq 0 \quad \text{and} \quad \mu \geq 1,$$

which express that the origin must lie in a closed cone spanned by the elements of $\mathcal{A}$ and $\{1_\omega : \omega \in \Omega\}$.

A typical inference drawn from a set of almost desirable gambles $\mathcal{A} \Subset \mathcal{L}$ (or the lower prevision it may be derived from) is the lower prevision for a gamble $f \in \mathcal{L}$. This is calculated using *natural extension* [8, §3.1], i.e., the linear program below:

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

By introducing variables $\mu \in \mathbb{R}^{\Omega}$, the constraints can be written as

$$\textstyle\sum_{g \in \mathcal{A}} \lambda_g \cdot g + \sum_{\omega \in \Omega} \mu_\omega \cdot 1_\omega + \alpha = f \quad \text{and} \quad \lambda \geq 0 \quad \text{and} \quad \mu \geq 0,$$

which express that f must lie in the closed cone spanned by the elements of $\mathcal{A}$, $\{1_\omega : \omega \in \Omega\}$, and $\{1_\Omega, -1_\Omega\}$. This problem is always feasible: take, for example, $\lambda = 0$, $\alpha = \min f$, and $\mu = f - \min f$; it will be unbounded if $\mathcal{A}$ incurs sure loss.

On top of avoiding sure loss, a lower prevision $\underline{P} \in [\mathcal{K} \to \mathbb{R}]$, with $\mathcal{K} \Subset \mathcal{L}$, may be required to be *coherent* [8, §2.5]. This can be checked by verifying that for all $f \in \mathcal{L}$ the natural extension coincides with $\underline{P}f$, so we do not need to investigate this further.

Conditional Lower Previsions. Moving from unconditional to conditional lower previsions, the basic ideas stay the same, but become more involved because we need to take the conditioning events into account.

Avoiding partial loss [8, §7.1.2–3 & Notes 7.1(7.)] replaces avoiding sure loss. Checking whether a conditional lower prevision $\underline{P} \in [\mathcal{N} \to \mathbb{R}]$, with $\mathcal{N} \Subset \mathcal{L} \times \Omega^*$ non-empty, *incurs partial loss* amounts to solving the feasibility problem below [3, (17)], in which $\mathcal{B} := \{([h - \underline{P}(h|B)] \cdot 1_B, B) : (h, B) \in \mathcal{N}\}$:

$$\begin{aligned} &\text{find} && (\lambda, \varepsilon) \in \mathbb{R}^{\mathcal{B}} \times \mathbb{R}^{\mathcal{B}}, \\ &\text{subject to} && \textstyle\sum_{(g,B)\in\mathcal{B}} \lambda_{g,B} \cdot [g + \varepsilon_{g,B} \cdot 1_B] \leq 0 \quad \text{and} \quad \lambda > 0 \quad \text{and} \quad \varepsilon \gtrdot 0. \end{aligned}$$

An algorithm for solving this *bilinearly* constrained feasibility problem using a sequence of linear programs has been first presented by Walley et al. [10, Alg. 2] for a subclass and made explicit for the general case by Couso and Moral [3, Alg. 1]. By introducing variables $\nu \in \mathbb{R}^{\mathcal{B}} \times \mathbb{R}^{\mathcal{B}}$ and $\mu \in \mathbb{R}^{\Omega}$, the constraints can be written as

$$\begin{aligned} &\textstyle\sum_{(g,B)\in\mathcal{B}} \lambda_{g,B} \cdot [\nu_{g,B,g} \cdot g + \nu_{g,B,B} \cdot 1_B] + \sum_{\omega\in\Omega} \mu_\omega \cdot 1_\omega = 0 \\ &\text{and} \quad \lambda > 0 \quad \text{and} \quad \nu \gtrdot 0 \quad \text{and} \quad \mu \geq 0, \end{aligned}$$

which express that the origin must lie in a general cone spanned by elements of $\bigcup_{(g,B)\in\mathcal{B}} \{(1-\delta) \cdot g + \delta \cdot 1_B : 0 < \delta < 1\}$ and $\{1_\omega : \omega \in \Omega\}$.

Inferring the lower prevision of a gamble $f \in \mathcal{L}$ conditional on an event $C \subseteq \Omega$ from a given conditional lower prevision $\underline{P} \in [\mathcal{N} \to \mathbb{R}]$ as above is calculated using a generalization of the natural extension procedure seen before [3, (21)]:

$$\begin{aligned} &\text{maximize} && \alpha \in \mathbb{R}, \\ &\text{subject to} && f \cdot 1_C - \alpha \cdot 1_C \geq \textstyle\sum_{(g,B)\in\mathcal{B}} \lambda_{g,B} \cdot [g + \varepsilon_{g,B} \cdot 1_B] \quad \text{and} \quad \lambda \geq 0 \quad \text{and} \quad \varepsilon \gtrdot 0. \end{aligned}$$

Again, this problem can be solved using linear program iteration [10, Alg. 4]-[3, Alg. 2]. By introducing variables $\nu \in \mathbb{R}^{\mathcal{B}} \times \mathbb{R}^{\mathcal{B}}$ and $\mu \in \mathbb{R}^{\Omega}$, the constraints can be written as

$$\begin{aligned} &\textstyle\sum_{(g,B)\in\mathcal{B}} \lambda_{g,B} \cdot [\nu_{g,B,g} \cdot g + \nu_{g,B,B} \cdot 1_B] + \sum_{\omega\in\Omega} \mu_\omega \cdot 1_\omega + \alpha \cdot 1_C = f \cdot 1_C \\ &\text{and} \quad \lambda \geq 0 \quad \text{and} \quad \nu \gtrdot 0 \quad \text{and} \quad \mu \geq 0, \end{aligned}$$

which express that $f \cdot 1_C$ must lie in a general cone spanned by elements of $\bigcup_{(g,B)\in\mathcal{B}} \{(1-\delta) \cdot g + \delta \cdot 1_B : 0 < \delta < 1\}$, $\{1_\omega : \omega \in \Omega\}$, and $\{1_C, -1_C\}$. This is always feasible: take, e.g., $\lambda = 0$, $\alpha = \min(f \cdot 1_C)$, and $\mu = [f - \min(f \cdot 1_C)] \cdot 1_C$.

3 Representation & Problem Formulation

In the previous section, we presented four problems from the literature that can essentially be formulated in terms of (conditional) lower previsions and that result in *specific* general cones. General sets of desirable gambles [8, App. F]-[9, §6]-[4] take the form of far more *general* general cones, and in generalizations of desirability [6] essentially any general cone could appear. So we must be able to deal with the feasibility and optimization problems that arise out of working with such models.

In this section, we will first discuss a representation for finitary general cones. Then we use that representation to formulate the general problem we want to tackle.

Representation of General Cones. We use the idea of Couso and Moral [3, Thm. 13; the need for such a representation is illustrated in Examples 3 & 4] to represent a finitary general cone as a convex hull of a finite number of finitary *open* cones. Formally, given a finite set of finite sets of gambles $\mathcal{R} \Subset \mathcal{L}^*$, the corresponding cone is

$$\underline{\mathcal{R}} \coloneqq \{\textstyle\sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \cdot \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g : \lambda > 0, \nu \gtrdot 0\}.$$

Whereas ‘finitary’ has a well-known meaning for open and closed cones—pointwise strict and non-strict convex hulls of finite sets of rays—, it is a concept we have fixed for ajar and therefore general cones by choosing a representation. To justify this choice, we employ facets [11]—the closed cones that are a closed cone’s maximally dimensional non-trivial faces—to give an appealing polytope-theoretically flavored definition and then show that the chosen representation satisfies it:

Definition. *An ajar cone $\mathcal{C}$ is* finitary *iff its closure* $\mathrm{cl}\mathcal{C}$ *is finitary and the intersection of $\mathcal{C}$ with each of* $\mathrm{cl}\mathcal{C}$*’s facets is a finitary (open, closed, or ajar) cone.*

Facet recursion is bound to stop in a finite number of steps: open and closed cones are terminal, with each step the dimension strictly decreases, and $\varnothing$ is clopen.

Theorem. *$\underline{\mathcal{R}}$ is a finitary general cone for every $\mathcal{R} \Subset \mathcal{L}^*$.*

Proof. Because $\mathcal{R}$ and its elements have finite cardinality, $\mathrm{cl}\,\underline{\mathcal{R}}$ is finitary; also $\mathrm{cl}\,\underline{\mathcal{R}} = \mathrm{cl}\,\underline{\mathcal{E}}$, with $\mathcal{E} \coloneqq \bigcup\mathcal{R}$. Let $\mathcal{F}$ be one of its (finite number of) facets, let $\mathcal{S} \coloneqq \{\mathcal{D} \in \mathcal{R} : \mathcal{D} \subset \mathcal{F}\}$, and let $\underline{\mathcal{D}} \coloneqq \{\mathcal{D}\}$. Then $\underline{\mathcal{S}} = \underline{\mathcal{R}} \cap \mathcal{F}$, because $\underline{\mathcal{D}} \setminus \mathcal{F} \neq \varnothing$ implies $\underline{\mathcal{D}} \cap \mathcal{F} = \varnothing$ by definition of $\underline{\mathcal{D}}$ and the fact that $\mathcal{F}$ is a facet. The proof is complete by facet recursion (replacing $\mathcal{R}$ by $\mathcal{S}$ in the first sentence). □

The approach of the proof also allows us to construct a canonical *cone-in-facet* representation (cf. concept of *zero-layers* [2, Ch. 12]). We illustrate this in Figure 1.

Formulation of the General Problem. Given a general cone represented by $\mathcal{R} \Subset \mathcal{L}^*$ and a gamble $h \in \mathcal{L}$, we wish to

$$\begin{aligned}
&\text{find} && (\lambda, \nu) \in \mathbb{R}^{\mathcal{R}} \times \textstyle\bigtimes_{\mathcal{D}\in\mathcal{R}} \mathbb{R}^{\mathcal{D}} \\
&\text{or maximize an affine function of} && \mu \coloneqq (\lambda_{\mathcal{D}} \cdot \nu_{\mathcal{D},g} : \mathcal{D} \in \mathcal{R}, g \in \mathcal{D}), \\
&\text{subject to} && \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \lambda_{\mathcal{D}} \cdot \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g = h \\
&\text{and} && \lambda > 0 \quad \text{and} \quad \nu \gtrdot 0 \\
&\text{and} && \text{possibly other, linear constraints (POLC) on } \mu.
\end{aligned}$$

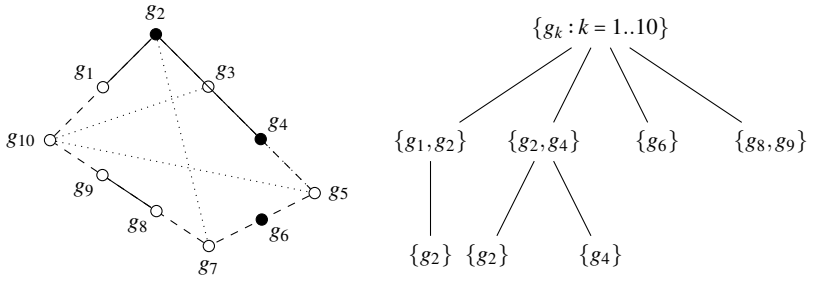

Fig. 1 On the left, we show the intersection of a cone $\underline{\mathcal{R}}$ with a plane. It can be represented by $\mathcal{R} \coloneqq \big\{\{g_3, g_5, g_{10}\}, \{g_1, g_2\}, \{g_2, g_7\}, \{g_8, g_9\}, \{g_2\}, \{g_4\}, \{g_6\}\big\}$. On the right, we order the sets $\mathcal{E}$—as defined in the proof of the Theorem—for each facet encountered in the facet recursion of $\mathcal{R}$ according to facet inclusion. Then $\big\{\{g_k : k = 1..10\}, \{g_1, g_2\}, \{g_2, g_4\}, \{g_6\}, \{g_8, g_9\}, \{g_2\}, \{g_4\}\big\}$ is the resulting cone-in-facet representation for $\underline{\mathcal{R}}$. Note that in terms of set sizes, this representation is not minimal, as $\big\{\{g_5, g_7, g_{10}\}, \{g_1, g_2\}, \{g_8, g_9\}, \{g_2\}, \{g_4\}, \{g_6\}\big\}$ also represents $\underline{\mathcal{R}}$.

Fig. 2 We use the CONEstrip algorithm to check whether g_3 and g_1 lie in the cone $\underline{\mathcal{R}}$ of Figure 1. First g_3: in the first iteration, $\tau_{\{g_2\}} = \tau_{\{g_4\}} = \tau_{\{-g_3\}} = 1$ and τ is zero for other indices; because possibly, e.g., $\mu_{\{g_3, g_5, g_{10}\}, g_3} > 0$, we might need a second iteration—with $\mathcal{S} = \big\{\{g_2\}, \{g_4\}, \{-g_3\}\big\}$—in which $\tau = 1$, so $g_3 \in \underline{\mathcal{R}}$. Next g_1: in the first iteration, $\tau_{\{g_2\}} = \tau_{\{g_1, g_2\}} = \tau_{\{-g_1\}} = 1$ and τ is zero for other indices; because necessarily $\mu_{\{g_3, g_5, g_{10}\}, g_{10}} > 0$, we need a second iteration—with $\mathcal{S} = \big\{\{g_2\}, \{g_1, g_2\}, \{-g_1\}\big\}$—which is infeasible, so $g_1 \notin \underline{\mathcal{R}}$.

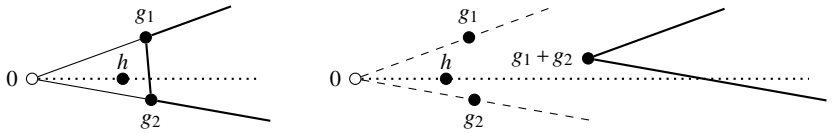

Fig. 3 On the left, we show the blunt closed cone $\mathcal{C} \coloneqq \{\lambda_1 \cdot g_1 + \lambda_2 \cdot g_2 : \lambda > 0\}$ (thin lines) and its proxy $\mathcal{C}' \coloneqq \{\mu_1 \cdot g_1 + \mu_2 \cdot g_2 : \mu \geq 0, \mu_1 + \mu_2 \geq 1\}$ (thick lines): $\{\kappa \cdot h : \kappa \geq 1\} \cap \mathcal{C}' \neq \varnothing$ is equivalent to $h \in \mathcal{C}$. On the right, we show the open cone $\mathcal{C} \coloneqq \{\nu_1 \cdot g_1 + \nu_2 \cdot g_2 : \nu > 0\}$ (dashed lines) and its proxy $\mathcal{C}' \coloneqq \{\mu_1 \cdot g_1 + \mu_2 \cdot g_2 : \mu \geq 1\}$ (thick lines): $\{\kappa \cdot h : \kappa \geq 1\} \cap \mathcal{C}' \neq \varnothing$ is equivalent to $h \in \mathcal{C}$.

To appreciate the general applicability of this problem for consistency checking and inference, let us look at how the problems discussed in Section 2 fit:

Incurring sure loss (given $\mathcal{A}$ of Sec. 2): $\mathcal{R} \coloneqq \big\{\{1_\omega : \omega \in \Omega\} \cup \mathcal{A} \setminus \{0\}\big\}$ and $h \coloneqq 0$.
Unconditional natural extension (given $\mathcal{A}$ and f of Sec. 2):
$\mathcal{R} \coloneqq \big\{\{g\} : g \in \mathcal{A}\big\} \cup \big\{\{1_\Omega\}, \{-1_\Omega\}, \{0\}\big\} \cup \big\{\{1_\omega\} : \omega \in \Omega\big\}$ and $h \coloneqq f$;
objective function expression $\mu_{\{1_\Omega\}, 1_\Omega} - \mu_{\{-1_\Omega\}, -1_\Omega}$.
Incurring partial loss (given $\mathcal{B}$ of Sec. 2):
$\mathcal{R} \coloneqq \big\{\{g, 1_B\} : (g, B) \in \mathcal{B} \setminus [\{0\} \times \Omega^*]\big\} \cup \big\{\{1_\omega\} : \omega \in \Omega\big\}$ and $h \coloneqq 0$.
Conditional natural extension (given $\mathcal{B}$, f, and C of Sec. 2):
$\mathcal{R} \coloneqq \big\{\{g, 1_B\} : (g, B) \in \mathcal{B}\big\} \cup \big\{\{1_C\}, \{-1_C\}, \{0\}\big\} \cup \big\{\{1_\omega\} : \omega \in \Omega\big\}$ and $h \coloneqq f \cdot 1_C$;
objective function expression $\mu_{\{1_C\}, 1_C} - \mu_{\{-1_C\}, -1_C}$.

The inclusion and exclusion of $\{0\}$ is essentially a way of selecting between constraints $\lambda \geq 0$ and $\lambda > 0$.

4 The CONEstrip Algorithm

Now that we have formulated the general problem and have a feel for the general cone representation used in this formulation, we are ready to work towards the actual algorithm that will allow us to solve this general problem. Before presenting the algorithm itself, we perform a supporting analysis of the general problem.

Analysis of the General Problem. In the general problem formulation we gave in Section 3, the (non-additional) constraints express that h must lie in the general cone represented by $\mathcal{R}$. For the feasibility problem, we can actually assume in all generality that $h = 0$, because we allow additional linear constraints on μ. To see this, consider the original feasibility problem and write it as

$$\begin{aligned}
\text{find}\quad & (\lambda, \nu) \in \mathbb{R}^{\mathcal{R}\cup\{\{-h\}\}} \times \bigtimes_{\mathcal{D}\in\mathcal{R}\cup\{\{-h\}\}} \mathbb{R}^{\mathcal{D}},\\
\text{subject to}\quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}\cup\{\{-h\}\}} \lambda_{\mathcal{D}} \cdot \sum_{g\in\mathcal{D}} \nu_{\mathcal{D},g} \cdot g = 0\\
\text{and}\quad & \lambda > 0 \quad\text{and}\quad \nu \gtrdot 0 \quad\text{and}\quad \mu_{\{-h\},-h} = \lambda_{\{-h\}} \cdot \nu_{\{-h\},-h} \geq 1\\
\text{and}\quad & \text{POLC on } \mu := (\lambda_{\mathcal{D}} \cdot \nu_{\mathcal{D},g} : \mathcal{D} \in \mathcal{R}, g \in \mathcal{D}),
\end{aligned}$$

whose feasible solutions μ can be related to those of the original problem by dividing them by $\mu_{\{-h\},-h}$.

For this feasibility problem, checking whether or not 0 lies in the *blunt* closure of $\underline{\mathcal{R}}$ can be done by solving the following linear programming feasibility problem:

$$\begin{aligned}
\text{find}\quad & \mu \in \bigtimes_{\mathcal{D}\in\mathcal{R}} \mathbb{R}^{\mathcal{D}},\\
\text{subject to}\quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = 0 \quad\text{and}\quad \mu \geq 0 \quad\text{and}\quad \text{POLC on } \mu\\
\text{and}\quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \geq 1.
\end{aligned}$$

where the last constraint is a proxy for the blunting—i.e., using $>$ instead of $\geq$—implied by the constraint $\lambda > 0$ (cf. Figure 3). If this problem is feasible, then 0 lies either in the interior of $\underline{\mathcal{R}}$ or in a facet. The interior case can be checked by solving another linear programming feasibility problem:

$$\begin{aligned}
\text{find}\quad & \mu \in \bigtimes_{\mathcal{D}\in\mathcal{R}} \mathbb{R}^{\mathcal{D}},\\
\text{subject to}\quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = 0 \quad\text{and}\quad \mu \geq 1 \quad\text{and}\quad \text{POLC on } \mu,
\end{aligned}$$

where $\mu \geq 1$ is a proxy for the constraint $\nu \gtrdot 0$ (cf. Figure 3).

Now, if h lies in the closure, but not the interior, it must lie on a facet. From the proof of the Theorem we know that we are then actually faced with the same type of feasibility problem we started out with, but now with $\mathcal{R}$ replaced by $\mathcal{S} \subset \mathcal{R}$. Based on this insight, we could construct a recursion algorithm. However, this would involve facet enumeration, which is a computationally expensive operation [1]. Nevertheless, the key insight is that, to solve the feasibility problem, we must identify the facet containing 0. Put differently, we must eliminate those elements from $\mathcal{R}$ that preclude 0 from lying in the (relative) interior of $\underline{\mathcal{R}}$.

The Algorithm for the Feasibility Problem. The idea is to relax the general feasibility problem to a linear programming problem and detect which elements $\mathcal{D}$ of $\mathcal{R}$ make the problems infeasible if $\lambda_{\mathcal{D}} > 0$. This is done by transforming it into a blunted closure-case optimization problem with an objective that essentially rewards solutions that are close to interior-case solutions. So, given the general feasibility problem of Section 3 with arbitrary $\mathcal{R} \Subset \mathcal{L}^*$ and $h \coloneqq 0$, the algorithm is:

1.
$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \tau_{\mathcal{D}}, \\ \text{subject to} \quad & \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = 0 \quad \text{and} \quad \mu \geq 0 \quad \text{and} \quad \text{POLC on } \mu \\ \text{and} \quad & 0 \leq \tau \leq 1 \quad \text{and} \quad \forall \mathcal{D} \in \mathcal{R} : \tau_{\mathcal{D}} \leq \mu_{\mathcal{D}} \quad \text{and} \quad \textstyle\sum_{\mathcal{D}\in\mathcal{R}} \tau_{\mathcal{D}} \geq 1. \end{aligned}$$

2. a. If there is no feasible solution, then the general problem is infeasible.
   b. Otherwise set $\mathcal{S} \coloneqq \{\mathcal{D} \in \mathcal{R} : \tau_{\mathcal{D}} > 0\}$; τ is equal to 1 on $\mathcal{S}$:
      i. If $\forall \mathcal{D} \in \mathcal{R} \setminus \mathcal{S} : \mu_{\mathcal{D}} = 0$, then the general problem is feasible.
      ii. Otherwise, return to step 1 with $\mathcal{R}$ replaced by $\mathcal{S}$.

We call this the *CONEstrip algorithm* because of step 2(b)ii, in which the irrelevant parts of the cone (representation) are stripped away. In Figure 2, the algorithm is illustrated on the cone of Figure 1. The algorithm is implemented in and tested with murasyp [5].

Proposition. *The claims made in the CONEstrip algorithm are veracious and it terminates after at most* $|\mathcal{R}| - 1$ *iterations.*

Proof. First the claim in step 2a: the feasibility requirements of the problem in step 1 are weaker than those of the general problem for the *current* representation $\mathcal{R}$ ($\sum_{\mathcal{D}\in\mathcal{R}} \tau_{\mathcal{D}} \geq 1$ is a proxy for $\lambda > 0$). Next the claim in step 2b: if μ is a solution, then $\mu / \min\{\tau_{\mathcal{D}} : \mathcal{D} \in \mathcal{S}\}$ is a solution for which the claim can be satisfied, which will be the case, because it increases the objective. Finally the claim in step 2(b)i: if the condition of the claim is verified, then μ is a solution to the general problem for the *current* representation $\mathcal{R}$ (take λ equal to 1 and ν equal to μ for indices in $\mathcal{S}$, and λ equal to 0 and ν arbitrary—e.g., 1—for other indices).

Now let $\mathcal{E} \coloneqq \{g \in \bigcup \mathcal{R} : (\exists \mathcal{D} \in \mathcal{R} : \mu_{\mathcal{D},g} > 0)\}$; by step 2(b)ii we know that $\underline{\mathcal{E}}$ contains 0. Moreover, $\underline{\mathcal{S}} = \underline{\mathcal{R}} \cap \underline{\mathcal{E}}$, so reiterating with $\mathcal{R}$ replaced by $\mathcal{S}$ leads to an equivalent problem feasibility-wise. Each iteration, by step 2(b)ii, we know that $|\mathcal{S}| < |\mathcal{R}|$, so at most $|\mathcal{R}| - 1$ iterations are necessary to decide feasibility of the original problem. □

The Algorithm for the Optimization Problem. The idea is to split off the nonlinear aspect of the feasibility part of the general optimization problem and deal with it using the CONEstrip algorithm. The optimization itself is then reduced to a linear programming problem. So, given the general optimization problem of Section 3 with arbitrary $\mathcal{R} \Subset \mathcal{L}^*$ and $h \in \mathcal{L}$, the proposed algorithm is:

1. Apply the CONEstrip algorithm to $\mathcal{R} \cup \{-h\}$ with $\mu_{\{-h\},-h} \geq 1$ as an additional constraint; if feasible, continue to the next step with the terminal set $\mathcal{S}$.
2.
$$\begin{aligned} \text{maximize an affine function of} \quad & \mu \in \textstyle\times_{\mathcal{D}\in\mathcal{R}} \mathbb{R}^{\mathcal{D}}, \\ \text{subject to} \quad & \textstyle\sum_{\mathcal{D}\in\mathcal{S}} \sum_{g\in\mathcal{D}} \mu_{\mathcal{D},g} \cdot g = h \\ \text{and} \quad & \mu \geq 0 \quad \text{and} \quad \text{POLC on } \mu, \end{aligned}$$

   where $\mu \geq 0$ is a proxy for $\mu \gtrdot 0$ by continuity of the linear objective.

5 Conclusion

We now have an efficient, polynomial time algorithm for consistency checking and inference in uncertainty modeling frameworks using general cones: the number of linear programs to solve has worst-case complexity linear in the cardinality of the cone representation. The work of Walley et al. [10] made me believe such an efficient algorithm was possible, the representation of Couso and Moral [3] provided useful structure, and a variable-bounding technique spotted in the 'zero norm'-minimization literature [7, (1) to (2)] made everything come together.

The CONEstrip algorithm—formulated in terms of linear programs—is rather high-level. Integrating it with a specific linear programming solver might allow for a practical increase in efficiency: e.g., the stripping step can be seen as a form of column elimination. Also, heuristics could be found to reduce the representation size.

The question may arise whether the representation and the algorithm are also applicable when modeling uncertainty using general bounded polytopes, such as non-closed credal sets (arising, e.g., when strict bounds on expectations are allowed). Yes: such polytopes can be seen as intersections of a general cone and a hyperplane.

Acknowledgements For useful discussion, I thank Dirk Aeyels, Gert de Cooman, Nathan Huntley, and especially Filip Hermans, who also provided very helpful pointers and feedback. I thank the reviewers for their effort and a much appreciated critical reading.

References

1. Avis D, Bremner D, Seidel R (1997) How good are convex hull algorithms? Computational Geometry 7:265–301, DOI 10.1016/S0925-7721(96)00023-5
2. Coletti G, Scozzafava R (2002) Probabilistic logic in a coherent setting. Kluwer Academic Publishers, DOI 10.1007/978-94-010-0474-9
3. Couso I, Moral S (2011) Sets of desirable gambles: conditioning, representation, and precise probabilities. International Journal of Approximate Reasoning 52(7):1034–1055, DOI 10.1016/j.ijar.2011.04.004
4. Quaeghebeur E (at the editor) Desirability. In: Coolen FPA, Augustin T, De Cooman G, Troffaes MCM (eds) Introduction to Imprecise Probabilities, Wiley
5. Quaeghebeur E (in progress) murasyp: Python software for accept/reject statement-based uncertainty modeling. URL http://equaeghe.github.com/murasyp
6. Quaeghebeur E, De Cooman G, Hermans F (in preparation) Accept & reject statement-based uncertainty models
7. Rinaldi F, Schoen F, Sciandrone M (2010) Concave programming for minimizing the zero-norm over polyhedral sets. Computational Optimization and Applications 46(3):467–486, DOI 10.1007/s10589-008-9202-9
8. Walley P (1991) Statistical Reasoning with Imprecise Probabilities. Chapman & Hall, London
9. Walley P (2000) Towards a unified theory of imprecise probability. International Journal of Approximate Reasoning 24(2-3):125–148, DOI 10.1016/S0888-613X(00)00031-1
10. Walley P, Pelessoni R, Vicig P (2004) Direct algorithms for checking consistency and making inferences from conditional probability assessments. Journal of Statistical Planning and Inference 126(1):119–151, DOI 10.1016/j.jspi.2003.09.005
11. Ziegler GM (1995) Lectures on Polytopes. Springer