Efficient Algorithms for Checking Avoiding Sure Loss

Nawapon Nakharutai NAWAPON.NAKHARUTAI@DURHAM.AC.UK
Matthias C. M. Troffaes MATTHIAS.TROFFAES@DURHAM.AC.UK
Camila C. S. Caiado C.C.D.S.CAIADO@DURHAM.AC.UK
Durham University
Durham (United Kingdom)

Abstract

Sets of desirable gambles provide a general representation of uncertainty which can handle partial information in a more robust way than precise probabilities. Here we study the effectiveness of linear programming algorithms for determining whether or not a given set of desirable gambles avoids sure loss (i.e. is consistent). We also suggest improvements to these algorithms specifically for checking avoiding sure loss. By exploiting the structure of the problem, (i) we slightly reduce its dimension, (ii) we propose an extra stopping criterion based on its degenerate structure, and (iii) we show that one can directly calculate feasible starting points in various cases, therefore reducing the effort required in the presolve phase of some of these algorithms. To assess our results, we compare the impact of these improvements on the simplex method and two interior point methods (affine scaling and primal-dual) on randomly generated sets of desirable gambles that either avoid or do not avoid sure loss. We find that the simplex method is outperformed by the primal-dual and affine scaling methods, except for very small problems. We also find that using our starting feasible point and extra stopping criterion considerably improves the performance of the primal-dual and affine scaling methods.

Keywords: avoiding sure loss; linear programming; benchmarking; simplex method; affine scaling method; primal-dual method; algorithm.

1. Introduction

Consider a subject modelling uncertainty about an experiment. One way of doing so, is by using gambles. A gamble is a (for instance, monetary) transaction that depends on the outcome of the experiment. To express her beliefs about the outcomes of the experiment, the subject can simply state which gambles she would accept, instead of directly specifying probabilities for the outcomes. A set of gambles that are considered acceptable to a subject is called a *set of desirable gambles*.

Williams (1975, 2007) was the first to give a full axiomatic treatment for sets of desirable gambles, and formalized a consistency principle called *avoiding sure loss*. This principle dictates that no combination of acceptable gambles should lead to a certain loss. Walley (1991, p.175) proposed a linear programming problem for checking avoiding sure loss, which was further studied and extended by various authors (Walley et al., 2004; Quaeghebeur, 2014). These linear programming problems can be solved by methods such as the primal-dual, simplex and affine scaling methods.

In the literature, to the best of our knowledge, there has been little to no discussion about which algorithm should be used to solve linear programming problems for avoiding sure loss. Walley (1991, p. 511) suggested that Karmarkar’s method may be useful for solving large problems.

The main contribution of this paper is a comparative study and analysis of how we can solve linear programs for avoiding sure loss most effectively, by looking at the three methods mentioned above. We exploit the structure of this program and also the interactions between the structure and the details of the algorithms. Specifically, we slightly reduce its dimension. We also propose an extra stopping criterion based on its degenerate structure. Finally, we show that one can directly calculate feasible starting points in various cases, therefore reducing the effort required in the presolve phase of some of these algorithms. Through a simulation study, we compare the impact of these improvements in the simplex method and two interior point methods (affine scaling and primal-dual) on randomly generated sets of desirable gambles that either avoid or do not avoid sure loss.

This paper is organised as follows. After reviewing linear programming problems and avoiding sure loss in Section 2, we give a linear programming problem and its dual for checking avoiding sure loss in Section 3. In Sections 4 to 6, we give a brief outline of primal-dual, simplex and affine scaling methods. We also study the different benefits of these three methods with respect to checking for avoiding sure loss. Algorithms for generating random sets of desirable gambles are outlined in Section 7 followed by a comparison of the efficiency of the methods. Section 8 concludes the paper.

2. Preliminaries

2.1 Linear programming problems

A linear programming problem is a problem of optimising a linear function (objective function) subject to constraints of linear equalities and linear inequalities. Because maximising a linear function is equivalent to minimising that function with a sign change, and because any linear inequality can be rewritten as a linear equality by adding non-negative slack variables, we have that every linear programming problem can be formulated as follows:

$$\min \mathbf{c}^\intercal \mathbf{x} \qquad \text{subject to} \qquad \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}, \tag{P}$$

where $A \in \mathbb{R}^{m \times n}$ with rank m and $m \leq n$. We call Eq. (P) the *primal* problem. The dual of Eq. (P) is:

$$\max \mathbf{b}^\intercal \mathbf{y} \qquad \text{subject to} \qquad \mathbf{A}^\intercal \mathbf{y} + \mathbf{t} = \mathbf{c} \text{ and } \mathbf{t} \geq \mathbf{0} \quad (\mathbf{y} \text{ free}). \tag{D}$$

We can solve either the primal or the dual problem because they have the same solution.

A solution that satisfies all constraints is called a *feasible solution*. An *optimal solution* is a feasible solution that achieves the optimal value of the objective function. A *basic* feasible solution is a feasible solution with at most m non-zero variables. It can be shown that every basic feasible solution is an extreme point, and vice versa (Fang and Puthenpura, 1993).

A linear programming problem is *degenerate* when it has basic feasible solutions in which fewer than m variables are non-zero. As we shall see, one way to check avoiding sure loss is by solving a degenerate linear programming problem where $\mathbf{b} = \mathbf{0}$. In that case, the following lemma is helpful. It is a generalised version of an exercise in Vanderbei (2001, p. 42, exercise 3.4).

Lemma 1 *The linear programming problem* $\min \mathbf{c}^\intercal \mathbf{x}$ *subject to* $\mathbf{A}\mathbf{x} \geq \mathbf{0}$ *either has an optimal value that is zero, or is unbounded.*

This lemma is very useful since it tells us that as soon as we find a feasible solution with a negative objective function value, then the problem is unbounded.

2.2 Avoiding sure loss

Throughout this paper, let Ω be a finite set of uncertain outcomes. A gamble is a bounded real-valued function on Ω. We think of a gamble as an uncertain reward expressed in units of utility.

Let $\mathcal{D}$ be a finite set of gambles that a subject is willing to accept; we refer to $\mathcal{D}$ as the subject's set of desirable gambles. The desirability axioms state that a non-negative combination of desirable gambles should not produce a sure loss (Walley, 1991, p.151). In that case, we say that $\mathcal{D}$ avoids sure loss. Formally:

Definition 2 *(Walley, 1991, p.151) The set* $\mathcal{D} = \{f_1, \dots, f_n\}$ *is said to avoid sure loss if for all* $n \in \mathbb{N}$, $\lambda_1, ..., \lambda_n \geq 0$ *and* $f_1, ..., f_n \in \mathcal{D}$*:*

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

3. Linear programs for checking avoiding sure loss

In this section, we study linear programming problems to check avoiding sure loss. The following linear programming problem is given by Walley (1991, p.175):

Theorem 3 *(Walley, 1991, p.175) The set* $\mathcal{D} = \{f_1, \dots, f_n\}$ *avoids sure loss if and only if the optimal value of the following linear programming problem is zero:*

$$\begin{aligned} \textit{(A1)} \qquad & \min \quad \alpha & (2)\\ \textit{subject to} \qquad & \forall\omega\in\Omega : \sum_{i=1}^{n}\lambda_i f_i(\omega) \leq \alpha & (3)\\ \textit{where} \qquad & \lambda_i \geq 0. & (4) \end{aligned}$$

We propose an alternative linear programming problem, which is slightly smaller in size, and which has only non-negative variables:

Theorem 4 *(Nakharutai, 2015, p.32) Choose any* $\omega^0 \in \Omega$*. The set* $\mathcal{D} = \{f_1, \dots, f_n\}$ *avoids sure loss if and only if the optimal value of the following linear programming problem is zero:*

$$\begin{aligned} \textit{(A2)} \qquad & \min \quad \sum_{i=1}^{n}\lambda_i f_i(\omega^0) + \alpha & (5)\\ \textit{subject to} \qquad & \forall\omega\in\Omega\setminus\{\omega^0\} : \sum_{i=1}^{n}\lambda_i(f_i(\omega^0) - f_i(\omega)) + \alpha \geq 0 & (6)\\ \textit{where} \qquad & \lambda_i, \alpha \geq 0. & (7) \end{aligned}$$

Note that (A1) and (A2) are fully degenerate because their right hand side is zero. Therefore, Theorem 1 applies to both of these problems.

When solving linear programs, typical algorithms such as primal-dual, simplex and affine scaling, require all free variables to be rewritten as a difference of two non-negative variables. So (A1) needs the introduction of an extra variable due to the presence of a free variable. Moreover, (A2) already has one fewer variable than (A1). Therefore, solving (A2) is easier than (A1).

Let's consider the dual of the problem (A2).

Theorem 5 *(Nakharutai, 2015, p.49) Choose any $\omega^0 \in \Omega$. The set $\mathcal{D} = \{f_1, \dots, f_n\}$ avoids sure loss if and only if the following linear programming problem has a feasible solution.*

$$
\begin{aligned}
\textit{(B1)} \quad & \max \quad 0 && (8)\\
\textit{subject to} \quad & \forall f_j \in \mathcal{D} : \sum_{\omega \in \Omega \setminus \{\omega^0\}} (f_j(\omega^0) - f_j(\omega)) p(\omega) \leq f_j(\omega^0) && (9)\\
& \sum_{\omega \in \Omega \setminus \{\omega^0\}} p(\omega) \leq 1 && (10)\\
\textit{where} \quad & p(\omega) \geq 0. && (11)
\end{aligned}
$$

How should we choose ω^0? Looking at the primal problem (A2), it is not obvious which ω^0 we should choose. However, since optimality in the primal problem corresponds to feasibility in the dual problem (Goh and X.Q.Yang, 2002, p.104), if we choose an ω^0 for which most values $f_j(\omega^0)$ are non-negative, then we can start (B1) closer to a feasible solution in the dual, and therefore closer to an optimal solution in the primal. For example, if there is an ω^0 for which $f_j(\omega^0) \geq 0$ for all j, then we immediately find a feasible solution by setting $p(\omega) = 0$ for all $\omega \neq \omega^0$.

4. Checking avoiding sure loss using primal-dual methods

4.1 Primal-dual methods

The primal-dual method solves the primal and dual problems simultaneously, see Fang and Puthenpura (1993); Griva et al. (2009) for more detail. It finds an optimal primal-dual solution $[\mathbf{x}^* \ \mathbf{y}^* \ \mathbf{t}^*]$ by repeatedly solving the following equalities:

$$
\begin{bmatrix} \mathbf{A}\mathbf{x} - \mathbf{b} \\ \mathbf{A}^\intercal \mathbf{y} + \mathbf{t} - \mathbf{c} \\ \mathbf{x}^\intercal \mathbf{t} \end{bmatrix} = \mathbf{0} \qquad \text{subject to} \qquad \mathbf{x}, \mathbf{t} \geq \mathbf{0} \tag{12}
$$

whilst keeping the variables $\mathbf{x}$ and $\mathbf{t}$ positive. The algorithm will stop when the primal residual $\mathbf{A}\mathbf{x} - \mathbf{b}$, dual residual $\mathbf{A}^\intercal \mathbf{y} + \mathbf{t} - \mathbf{c}$ and duality gap $\mathbf{x}^\intercal \mathbf{t}$ are small enough or when an unboundedness criterion is satisfied in either the primal or the dual problems (Fang and Puthenpura, 1993).

Theoretically, the primal-dual method generates iterates $\mathbf{x},\ \mathbf{y},\ \mathbf{t}$, where $\mathbf{x},\ \mathbf{t} > 0$, that stay in the feasible region. However, in practical implementations, keeping $\mathbf{x},\ \mathbf{y},\ \mathbf{t}$ in the feasible region is very difficult because of numerical problems. Therefore, practical implementations of primal-dual algorithms start with an arbitrary point $[\mathbf{x}\ \mathbf{y}\ \mathbf{t}]$ where $\mathbf{x},\ \mathbf{t} > \mathbf{0}$ and generate iterate points that converge to an optimal solution. Even though there is no convergence proof, this approach seems to work well in practice (Fang and Puthenpura, 1993).

4.2 Corresponding linear programming problems

The primal problem (A2) can be easily written as (P) by adding slack variables:

$$\text{(A3)} \qquad \min \quad \sum_{i=1}^{n} \lambda_i f_i(\omega^0) + \alpha \tag{13}$$

$$\text{subject to} \quad \forall \omega \in \Omega \setminus \{\omega^0\} : \sum_{i=1}^{n} \lambda_i (f_i(\omega^0) - f_i(\omega)) + \alpha - s(\omega) = 0 \tag{14}$$

$$\text{where} \quad \lambda_i,\ \alpha,\ s(\omega) \geq 0. \tag{15}$$

Note that these equalities are linearly independent due to the presence of distinct slack variables in each equality, so the corresponding matrix A has full rank. This will be the case for every system of equalities that we write down further as well; we will not repeat this point.

The dual problem (B1) can be written as (D) again by adding slack variables:

$$\text{(B2)} \qquad \max \quad 0 \tag{16}$$

$$\text{subject to} \quad \forall f_j \in \mathcal{D} : \sum_{\omega \in \Omega \setminus \{\omega^0\}} (f_j(\omega^0) - f_j(\omega)) p(\omega) + t_j = f_j(\omega^0) \tag{17}$$

$$\sum_{\omega \in \Omega \setminus \{\omega^0\}} p(\omega) + q = 1 \tag{18}$$

$$\text{where} \quad p(\omega),\ t_j,\ q \geq 0. \tag{19}$$

To apply the primal-dual method, (A3) and (B2) are easily written in the form of Eq. (12). Next, we will investigate the structure of the problems (A3) and (B2) to choose a suitable starting point.

4.3 Starting points and extra stopping criteria

For the dual problem (B2), a starting point can be $q^0 = p^0(\omega) = 1/|\Omega|$ for all $\omega \in \Omega \setminus \{\omega^0\}$, because these $p^0(\omega)$ and q^0 satisfy Eq. (18). For simplicity, we suggest choosing $t_j = 1$ for all j. Note that this solution may not be feasible (if it were, we would have solved the problem!).

For the primal problem (A3), we can find an initial interior feasible solution as follows:

Theorem 6 *An initial interior feasible solution of the specific linear programming problem*

$$\min \quad c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + \alpha \tag{20}$$

$$\textit{subject to} \quad \forall j = 1, \ldots, m : \sum_{k=1}^{n} a_{jk} x_k + \alpha - s_j = 0 \tag{21}$$

$$\textit{where} \quad \alpha \geq 0,\ x_k \geq 0 \textit{ and } s_j \geq 0. \tag{22}$$

is given by $x_k = 1$, $\alpha = 1 + \max\{0, -\delta\}$ *with* $\delta := \min_j \{\sum_{k=1}^{n} a_{jk}\}$ *and* $s_j = \alpha + \sum_{k=1}^{n} a_{jk}$.

Proof We must show that Eq. (21) is satisfied, and that all variables are strictly positive.

Clearly, Eq. (21) is satisfied by our choice of s_j, all $x_k = 1 > 0$, and $\alpha \geq 1 > 0$. Finally, note that also all $s_j > 0$ because

$$s_j = \alpha + \sum_{k=1}^{n} a_{jk} \geq \alpha + \delta \geq 1 - \delta + \delta > 0. \tag{23}$$

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

Note that for problem (A3), even though we start with a feasible solution, the solution does not necessarily remain feasible due to numerical rounding errors. In particular, we can apply our extra stopping criterion from Theorem 1 only if the primal residual $\mathbf{A}\mathbf{x} - \mathbf{b}$ is small.

In the next two sections, we look at the simplex and affine scaling methods, and we also suggest suitable linear programming representations.

5. Checking avoiding sure loss using simplex methods

5.1 Simplex methods and pivoting

The simplex method is an iterative method. At every iteration, we move from a current extreme point to another extreme point that decreases the objective function value. If a linear programming problem is in the form

$$\min\ \mathbf{c}^\mathsf{T}\mathbf{x} \qquad \text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{s} = \mathbf{b} \qquad \text{where} \ \ \mathbf{x}, \mathbf{s}, \mathbf{b} \geq 0, \tag{S}$$

then we immediately obtain a starting extreme point by setting $\mathbf{s} = \mathbf{b}$ and $\mathbf{x} = \mathbf{0}$.

A brief outline of the simplex method is given as follows (see Fang and Puthenpura (1993) for more detail). We first write (S) in the following format:

$$\begin{bmatrix} \mathbf{c}^\mathsf{T} & \mathbf{0}^\mathsf{T} & 0 \\ \mathbf{A} & \mathbf{I} & \mathbf{b} \end{bmatrix} \tag{24}$$

If there is a negative value in the top row, then we leave the current extreme point and move to an improved extreme point by performing a pivot via row operations. We repeat this until there is no negative value in the first row, or until we can no longer pivot.

5.2 Corresponding linear programming problems

Note that by multiplying Eq. (14) by -1, problem (A3) can be rewritten as (S). Since all the right hand side constraints are zero, there is only one extreme point that is $\mathbf{0}$. Consequently, $\mathbf{0}$ is the only extreme point. In this case either the optimal value is zero or the problem is unbounded.

Since the value of the objective function is always zero, we cannot apply Theorem 1. It is also worth noting that problem (A3) is degenerate and that the simplex method may cycle in such cases (Hoffman, 1953). Cycling can be detected by checking Bland's rule or Lexicographic Rule (Fang and Puthenpura, 1993, p.44) resulting in more calculations. Therefore, the simplex method may perform poorly when solving problem (A3).

We now look at the dual problem (B1). To convert (B1) into the standard form (S), if there exists $f_j(\omega^0) < 0$ for some j, then we multiply the corresponding constraint by -1 to make the right hand side non-negative and then add artificial variables. We obtain the following linear programming problem for which we can immediately give an initial extreme point.

Corollary 7 *The set $\mathcal{D} = \{f_1, \ldots, f_n\}$ avoids sure loss if and only if the optimal value of the following linear programming problem is zero.*

$$\begin{aligned}
\textit{(B3)} \quad & \min && \sum_{j\in N} t_j && (25)\\
& \textit{subject to} && \forall j \in N: \sum_{\omega\in\Omega\setminus\{\omega^0\}} (f_j(\omega) - f_j(\omega^0))p(\omega) - s_j + t_j = -f_j(\omega^0) && (26)\\
& && \forall j \in I \setminus N: \sum_{\omega\in\Omega\setminus\{\omega^0\}} (f_j(\omega^0) - f_j(\omega))p(\omega) + u_j = f_j(\omega^0) && (27)\\
& && \sum_{\omega\in\Omega\setminus\{\omega^0\}} p(x) + q = 1 && (28)\\
& \textit{where} && p(\omega), q, s_j, t_j, u_j \geq 0 && (29)
\end{aligned}$$

with $I \coloneqq \{1, \ldots, n\}$ and $N \coloneqq \{j \in N\colon f_j(\omega^0) < 0\}$.

We can then choose an initial extreme point to be $t_j = -f_j(\omega^0)$, $u_j = f_j(\omega^0)$, $q = 1$ and $p(\omega) = s_j = 0$.

Now, to check avoiding sure loss using the simplex method, we can solve either (A3) or (B3). If we want to avoid degeneracy and cycling, then we should solve (B3) rather than (A3).

6. Checking avoiding sure loss using affine scaling methods

6.1 Affine scaling methods

The idea of the affine scaling method is to generate a sequence of interior feasible solutions by repeatedly solving Eq. (P) such that the corresponding objective function values are decreasing. The assumption is that this sequence converges to the optimal solution if it exists. The method needs a starting interior feasible point, and we stop when the difference between objective function values is small enough or an unboundedness criterion is satisfied (see Fang and Puthenpura (1993); Griva et al. (2009); Saigal (1995) for more detail).

6.2 Extra stopping criteria

Recall that the right hand side constraints of (A3) are zero and the affine scaling method generates a sequence of interior feasible solutions. Therefore, we can apply Theorem 1. Specifically, the method can stop as soon as it finds a feasible solution with a negative objective function value. In this case, the problem is unbounded.

6.3 Initial feasible interior points

In practical implementations, there are several mechanisms for finding initial interior feasible points which require solving another linear programming problem. For the problem (A3), we do not need to use such mechanisms since we can apply Theorem 6 to find an initial interior feasible point. For the problem (B2), we can apply the following mechanism to obtain a starting interior feasible point (Fang and Puthenpura, 1993).

Given constraints $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$, we choose any $\mathbf{x}^0 > \mathbf{0}$ and calculate $\mathbf{y} = \mathbf{b} - \mathbf{A}\mathbf{x}^0$. If $\mathbf{y} = \mathbf{0}$, then $\mathbf{x}^0$ is an interior feasible solution. Otherwise, we solve

$$\min \gamma \qquad \text{subject to} \qquad \mathbf{Ax} + \mathbf{y}\gamma = \mathbf{b} \qquad \text{where } \mathbf{x} \geq \mathbf{0}, \gamma \geq 0 \tag{30}$$

by the affine scaling method with an interior feasible solution $[\mathbf{x} \;\; \gamma] = [\mathbf{x}^0 \;\; 1]$. If the affine scaling method finds an optimal solution $[\mathbf{x}^* \;\; \gamma^*]$ such that $\gamma^* = 0$, then $\mathbf{x}^*$ is an interior feasible solution of the original problem.

Let's write (B2) in the explicit form of Eq. (30). Let $\Omega \setminus \{\omega^0\} = \{\omega^1, ..., \omega^m\}$, we choose $\mathbf{x}^0 = \left[p^0(\omega_1) \;\; ... \;\; p^0(\omega_m) \;\; t_1^0 \;\; ... \;\; t_n^0 \;\; q^0\right] > \mathbf{0}$ and calculate $\mathbf{y} := \mathbf{b} - \mathbf{A}\mathbf{x}^0$ as follows:

$$y_j \coloneqq f_j(\omega^0) - \sum_{\omega \in \Omega \setminus \{\omega^0\}} (f_j(\omega^0) - f_j(\omega))p^0(\omega) - t_j^0 \tag{31}$$

$$z \coloneqq 1 - \left(\sum_{\omega \in \Omega \setminus \{\omega^0\}} p^0(\omega) + q^0 \right) \tag{32}$$

Note that we can choose $t_j^0 = 1$ for simplicity. We also choose $q^0 = p^0(\omega) = 1/|\Omega|$ for all $\omega \in \Omega \setminus \{\omega^0\}$ so that $z = 0$.

We then solve the following problem.

$$\text{(B4)} \qquad \min \qquad \gamma \tag{33}$$

$$\text{subject to} \qquad \forall f_j \in \mathcal{D} : \sum_{\omega \in \Omega / \{\omega^0\}} (f_j(\omega^0) - f_j(\omega))p(\omega) + t_j - y_j\gamma = f_j(\omega^0) \tag{34}$$

$$\sum_{\omega \in \Omega / \{\omega^0\}} p(\omega) + q = 1 \tag{35}$$

where $\mathbf{x}^0$ is an interior feasible point of (B4). After obtaining an optimal solution for (B4), if $\gamma^* = 0$, then $\mathbf{x}^*$ is an interior feasible solution for (B2) (and therefore also an optimal solution for (B2)); otherwise, there is no feasible solution.

7. Algorithms and numerical results

7.1 Algorithms for generating random sets of gambles

We give two algorithms for generating random sets of desirable gambles $\mathcal{D}$ that either avoid or do not avoid sure loss. In this section, Ω denotes the set of outcomes, $\Delta(\Omega)$ denotes the unit simplex over Ω and $\mathcal{D}$ denotes a set of desirable gambles. Algorithm 1 generates a random set of desirable gambles that avoids sure loss. Starting from a $\mathcal{D}$ that avoids sure loss, Algorithm 2 generates another gamble that can be added to violate consistency, thereby generating a set of desirable gambles that does not avoid sure loss.

7.2 Numerical results

We solve problem (A3) for checking avoiding sure loss by three methods. The simplex and primal-dual methods are available in MATLAB, while the affine scaling method is not. Unfortunately, the MATLAB implementation of the primal-dual method does not allow us to specify the initial

Algorithm 1: Generate a random set of desirable gambles $\mathcal{D}$ that avoids sure loss

Input : Number of gambles $J := |\mathcal{D}|$
Number of outcomes $|\Omega|$
Number of probability mass functions k

Output: A set of desirable gambles $\mathcal{D}$ that avoids sure loss

Stage 1. For each $i = 1 : k$, sample a single p_i uniformly from the unit simplex $\Delta(\Omega)$ as follows:

(a) For each ω, sample $q_i(\omega)$ uniformly from $(0, 1)$.
(b) For each ω, set $p_i(\omega) := (-\ln q_i(\omega)) \,/\, (-\sum_\omega \ln q_i(\omega))$.

Stage 2. Generate a set of desirable gambles $\mathcal{D}$

(a) For each ω and j, sample $g_j(\omega)$ uniformly from $(0, 1)$.
(b) For each j, set $\underline{P}(g_j) := \min_{i=1}^k \sum_\omega p_i(\omega) g_j(\omega)$.
(c) Set $\mathcal{D} := \{g_j - \underline{P}(g_j); j \in J\}$.

Algorithm 2: Generate a random set of desirable gambles $\mathcal{D}$ that does not avoid sure loss

Input : A set of desirable gambles $\mathcal{E} = \{f_1, \dots, f_J\}$ that avoids sure loss
$\delta > 0$

Output: A set of desirable gambles $\mathcal{D}$ that does not avoid sure loss

(a) For each ω, sample $g(\omega)$ uniformly from $(0, 1)$.

(b) Solve the following linear program:

$$\text{(C)} \qquad \min \quad \beta$$
$$\text{subject to} \quad \forall \omega \in \Omega : \sum_{j=1}^{J} \lambda_j f_j(\omega) - \beta \leq -g(\omega)$$
$$\text{where} \quad \lambda_i \geq 0 \;\; (\beta \text{ free}).$$

(c) Set $\underline{P}(g) := \beta + \delta$.

(d) Set $\mathcal{D} := \mathcal{E} \cup \{g - \underline{P}(g)\}$.

starting point and to add an extra stopping criterion (Theorem 1). To compare these three methods, we wrote our own implementation of the improved affine scaling and the improved primal-dual methods. Specifically, the extra stopping criterion and our method for the initial interior feasible point were included in our implementation of these two algorithms.

We generate two types of random sets of desirable gambles. For each type, we consider the scenarios $|\mathcal{D}| = J = 2^i$ for $i \in \{1, 2, \ldots, 8\}$ and $|\Omega| = 2^j$ for $j \in \{1, 2, \ldots, 8\}$. We also fixed $k = 2^4$; varying k had little impact on the results. We first generate a set that avoids sure loss using Algorithm 1. Next, we generate a set that does not avoid sure loss using Algorithm 2 with $\delta = 0.05$ and with $\mathcal{E}$ provided by Algorithm 1. We then benchmark the primal-dual, simplex and affine scaling methods by measuring their computational times applied to each generated set.

For each set of desirable gambles, we assume that we do not know whether it avoids sure loss or not, and pose the linear programming problem in the format of the primal problem (A3). We then solve each case using these three methods. 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 Fig. 1.

Figure 1 shows the average computational time spent during each method when checking avoiding sure loss. In the left column, the sets of desirable gambles avoid sure loss and in the right column, they do not avoid sure loss. Each row represents a different number of desirable gambles. The vertical axis represents the computational time. The horizontal axis shows the number of outcomes. 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 due to the sufficiently large sample size.

Overall, in the first two rows, where we compare the three methods, the simplex method is always outperformed by the improved primal-dual and the improved affine scaling methods. Regardless of whether we avoid sure loss or not, the improved primal dual method is faster than the improved affine scaling method except when we do not avoid sure loss and the number of desirable gambles is small.

In the last two rows, we compare our suggested improvements on the primal-dual method. When we avoid sure loss, using our feasible starting point shows a very slight improvement, although it is barely noticeable. The extra stopping criterion does not help at all in this case, quite logically so, because it will never be invoked in this case. What is important here is that it also does not hinder performance; the overhead of the extra check is thus negligible. When we do not avoid sure loss, both the extra stopping criterion and the feasible starting point considerably improve performance. Using both improvements gives the best results.

8. Conclusion

We studied and improved methods to solve linear programming problems efficiently for checking avoiding sure loss. By exploiting the structure of the linear programming problem, we first slightly reduced its dimension. Secondly, we proposed an extra stopping criterion based on its degenerate structure. We also showed that one can directly calculate feasible starting points in various cases, therefore reducing the effort required in the presolve phase of some of these algorithms.

We compared the impact of these improvements on linear programming methods (simplex, affine scaling, and primal-dual) on randomly generated sets of desirable gambles that either avoid

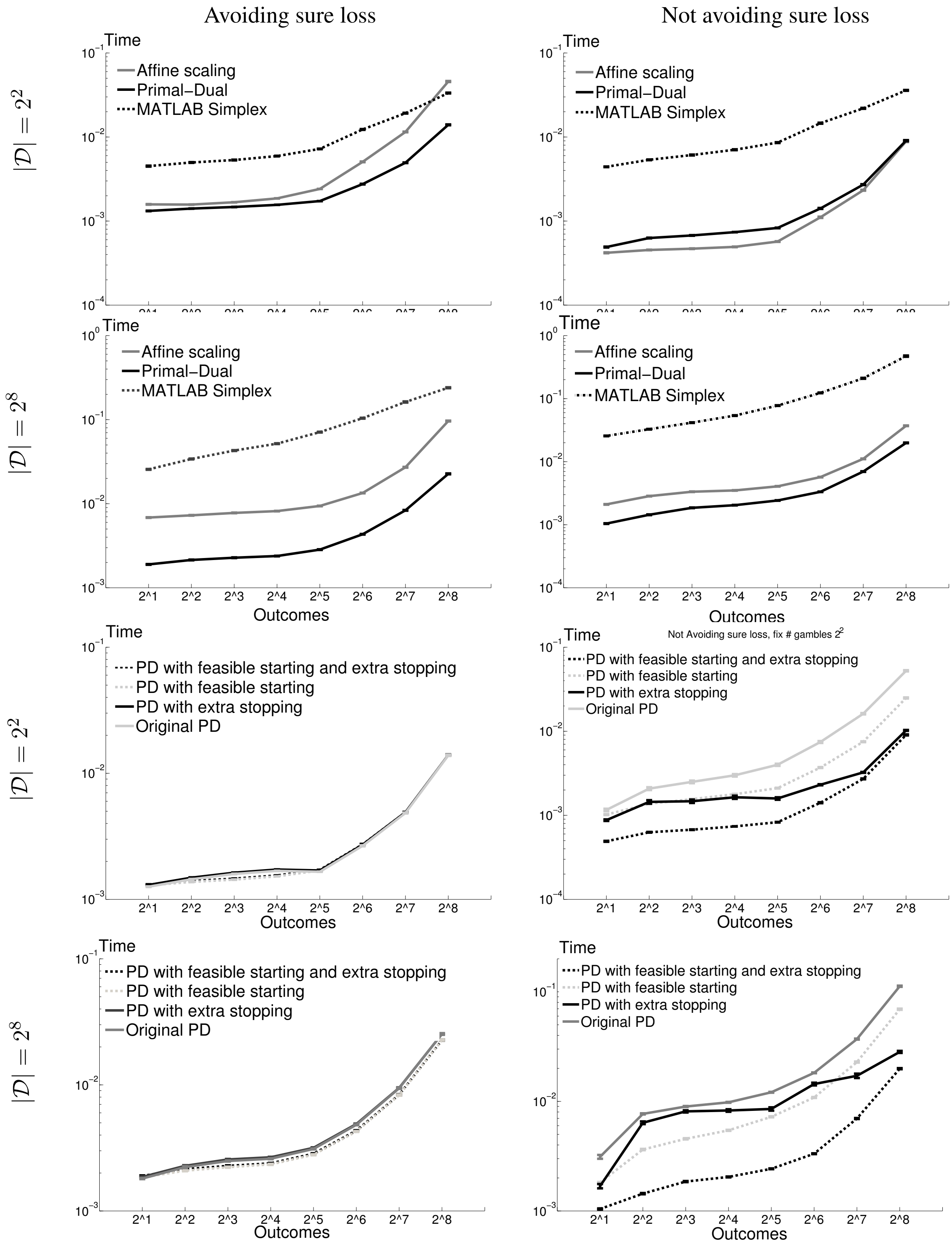

Figure 1: Comparison plots of the average computational time for three methods and for different improved primal-dual methods (PD).

or do not avoid sure loss. In our simulation, we found that the improved primal-dual and improved affine scaling methods outperform the simplex method. We found that both affine scaling and primal-dual methods benefit from the extra stopping criterion and feasible starting points. Overall, the improved primal-dual method is faster than the improved affine scaling method except when we do not avoid sure loss and the number of desirable gambles is small.

In future work, we will explore suitable starting points for the dual problems, algorithms for choosing ω^0 for large problems, and different structures for the credal set.

Acknowledgments

We would like to acknowledge support for this project from Development and Promotion of Science and Technology Talents Project (Royal Government of Thailand scholarship).

References

S.-C. Fang and S. Puthenpura. *Linear Optimization and Extensions: Theory and Algorithms*. Springer Science+Business Media New York, 1993.

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

I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization Second edition*. SIAM, Philadelphia, 2009.

A. J. Hoffman. Cycling in the simplex algorithm. *National Bureau of Standards, Washington, D.C*, 1953.

N. Nakharutai. Computing with lower previsions using linear programming. Master's thesis, Durham University, 2015. Unpublished thesis.

E. Quaeghebeur. *A Propositional CONEstrip Algorithm*, pages 466–475. Springer International Publishing, 2014. ISBN 978-3-319-08852-5. doi:10.1007/978-3-319-08852-5_48.

R. Saigal. *Linear programming : a modem integrated analysis*. Springer Science+Business Media New York, 1995.

R. J. Vanderbei. *Linear Programming: Foundations and Extensions, Second edition*. Springer, 2001.

P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

P. Walley, R. Pelessoni, and P. Vicig. Direct algorithms for checking consistency and making inferences from conditional probability assessments. *Journal of Statistical Planning and Inference*, 126:119–151, 2004.

P. M. Williams. Notes on conditional previsions. Technical report, School of Math. and Phys. Sci., Univ. of Sussex, 1975.

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