Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

SIMULATING TAIL EVENTS WITH UNSPECIFIED TAIL MODELS

Henry Lam

Department of Industrial and Operations Engineering
University of Michigan
1205 Beal Ave.
Ann Arbor, MI 48109, USA

Clementine Mottet

Department of Mathematics and Statistics
Boston University
111 Cummington Mall
Boston, MA 02215, USA

ABSTRACT

Reliable simulation estimation builds on accurately specified input models. In the context of simulating tail events, knowledge on the tail of the input model is especially important, yet is often hard to obtain due to a lack of data. In this paper, we consider tail event estimation without any knowledge on the input tail, but rather only making a general assumption that it is convex. We focus on the standard problem of estimating the probability for i.i.d. sum, and set out goal as to compute its worst-case bound among all summand distributions that have convex tails. Our main procedure relies on a stochastic, and in a sense infinite-dimensional, version of the Frank-Wolfe method in nonlinear programming. We demonstrate through a numerical example how the level of knowledge on the tail of the summands relates to the conservativeness in computing bounds for the aggregate tail quantity.

1 INTRODUCTION

This paper explores simulation-based estimation of probabilities related to tail events without full knowledge of the input model, especially its tail. The motivation is that, in most practical scenarios, information on the tail of an input model is difficult to obtain because of the inavailability of data. Specifically, we shall consider the standard problem of simulating the tail probability of i.i.d. sum, but each summand random variable has a distribution that is known only up to a certain threshold, but nothing beyond that. Our goal is to construct a methodology that can output reasonable estimate for the tail probability under such uncertainty.

The problem we post is generally ill-posed when no additional assumption is imposed, in the sense that the tail probability estimate of the i.i.d. sum can be anywhere from 0 to 1. Our first step to avoid such issue is to make a geometric assumption: the tail density of the i.i.d. random variables is convex. This assumption is clearly satisfied by any known parametric distributions, including normal, exponential, lognormal, Weibull etc., as well as distributions that are more pertinent to tail events like the generalized extreme value distribution, and therefore we believe it is a reasonable assumption to make.

Our approach is a worst-case analysis: imposing convexity as a constraint, we aim to find the most extreme tail probability of i.i.d. sum. This requires solving an optimization program that has decision variable being a probability density. This optimization has a nonlinear objective function and is infinite-dimensional. Our main contribution of the paper is to demonstrate a stochastic version of the Frank-Wolfe (FW) method that is well-suited for the problem, including being able to run in the infinite-dimensional space by exploiting the problem structure. In fact, due to this structure, our algorithm will operate on a space of what we call "augmented" probability densities, which is designed to handle sequences of densities that are not tight and have masses escaping to infinity. In overall, the implementability of the algorithm will provide a tractable way to obtain robust estimate of tail probabilities when the input model's tail is not fully known.

We close this introduction by discussing some related work. Our worst-case framework is inspired by the literature on distributionally robust optimization (Delage and Ye 2010, Ben-Tal et al. 2013), which considers decision-making in stochastic environment where the underlying probability is not fully known and is assumed to lie in an uncertainty set. There is also literature on so-called moment problems and Chebyshev's inequalities, applied to stochastic programming and decision analysis (Birge and Wets 1987, Smith 1995). In contrast to these literature, we impose tail convexity constraint, studied in Lam and Mottet (2015) and is also related to the general framework of moment problems in distributional class in Popescu (2005). References on the FW method can be found in Bertsekas (1999), and our stochastic version of FW is related to Ghosh and Lam (2015) and Goeva et al. (2014), which consider optimization over discretized input models as means for measuring model uncertainty or carrying out model calibration.

The paper is organized as follows. We present our formulation and notation in Section 2, and provide an overview of our procedure in Section 3. Then we discuss several aspects of the procedure, including the gradient form in Section 4, the solution to linearized subprograms in Section 5, and the construction of augmented probability densities in Section 6. We lay out the algorithmic details of our FW procedure in Section 7, followed by some numerics in Section 8. We discuss future work in Section 9.

2 FORMULATION

We are interested in estimating $P(S_n > b)$ where $S_n = X_1 + \cdots + X_n$, X_i's are i.i.d. random variables and b is some large number. We assume that X_i has density $f(x)$ for $x \in \mathbb{R}$ but is known only up to some threshold a. In other words, $f(x) = g(x)$ for some known function g for $x \le a$.

Beyond a, we assume that f is convex. We denote

$$\mathscr{A} = \{f(x), x \in \mathbb{R} : f(x) = g(x) \text{ for } x \le a,\ f(x) \text{ is convex for } x \ge a\} \tag{1}$$

as the set of densities that are equal to g for $x \le a$ and is convex beyond a. Our goal is to compute the worst-case tail probability

$$\max_{f \in \mathscr{A}} P_f(S_n > b) \tag{2}$$

where P_f denotes the probability measure generated by the i.i.d. random variables each with density f.

It is easy to check that the feasible region $\mathscr{A}$ is convex. However, the objective function in (2), which can be written as

$$P_f(S_n > b) = \int \cdots \int_{x_1 + \cdots + x_n > b} f(x_1) \cdots f(x_n) dx_1 \cdots dx_n \tag{3}$$

is not linear, nor convex with respect to f in general. In view of this, we shall set our target as to obtain a local optimum for (2).

3 MAIN ELEMENTS OF OUR PROCEDURE

The strategy that we shall undertake borrows from the FW method (Bertsekas 1999; also known as the conditional gradient method) in nonlinear programming. This method applies when the feasible region is convex. It is an iterative scheme that, at each iteration, linearizes the objective function and solves a subprogram with linear objective and the given feasible region in order to find a direction to move along. To explain further, let us denote $Z(f) = P_f(S_n > b)$. At each iteration k, given the current solution f_k, one would need to solve

$$\max_{f \in \mathscr{A}} \langle \nabla Z(f_k), f - f_k \rangle \tag{4}$$

where $\nabla Z(f)$ denotes some notion of the gradient of $Z(\cdot)$ with respect to f, and $\langle \cdot, \cdot \rangle$ is some suitable inner product. Say the optimal solution of (4) is r_k. Then one would update the solution $f_{k+1} = f_k + \varepsilon_k(r_k - f_k) = (1 - \varepsilon_k) f_k + \varepsilon_k r_k$ for some step size ε_k.

For our particular problem, it is difficult to evaluate the objective function or its gradient directly because this would involve high-dimensional convolution when n is large. So one has to resort to simulation estimation. Hence one would need to replace $\nabla Z(f_k)$ by some estimate $\widehat{\nabla Z(f_k)}$, and the corresponding stepwise subprogram will be

$$\max_{f\in\mathscr{A}} \langle \widehat{\nabla Z(f_k)}, f - f_k\rangle \tag{5}$$

with the update being $f_{k+1} = (1-\varepsilon_k)f_k + \varepsilon_k \hat{r}_k$ where $\hat{r}_k$ is the optimal solution of (5)

In the following sections, we will present the above notions and steps in detail.

4 LINEARIZATION OF OBJECTIVE FUNCTION

We first discuss the notion of gradient for $P_f(S_n > b)$ with respect to the density f. This is summarized as:

Lemma 1 For any probability densities f_1 and f_2, we have

$$P_{f_2}(S_n > b) \quad = \quad P_{f_1}(S_n > b) + \int_{-\infty}^{\infty} \zeta(x; f_1)(f_2(x) - f_1(x))dx + O(\|f_1 - f_2\|_1^2) \tag{6}$$

as $\|f_1 - f_2\|_1 \to 0$, where

$$\zeta(x; f_1) = nP_{f_1}(S_{n-1} > b - x)$$

and $\|f_1 - f_2\|_1 = \int_{-\infty}^{\infty} |f_1(x) - f_2(x)|dx$ denotes the L_1-distance between f_1 and f_2.

Proof of Lemma 1. The proof follows from an expansion

$$\begin{aligned}
& P_{f_2}(S_n > b) \\
= & \int \cdots \int_{\sum_{i=1}^n x_i > b} \prod_{i=1}^n f_2(x_i)dx_i \\
= & \int \cdots \int_{\sum_{i=1}^n x_i > b} \prod_{i=1}^n (f_1(x_i) + (f_2(x_i) - f_1(x_i)))dx_i \\
= & \int \cdots \int_{\sum_{i=1}^n x_i > b} \left(\prod_{i=1}^n f_1(x_i) + \sum_{i=1}^n \prod_{\substack{j=1,\ldots,n \\ j\neq i}} f_1(x_j)(f_2(x_i) - f_1(x_i)) \right. \\
& \left. + \sum_{r=2}^n \sum_{(i_1,\ldots,i_r)\in\mathscr{P}(n,r)} \prod_{\substack{j=1,\ldots,n \\ j\neq i_u, u=1,\ldots,r}} f_1(x_j) \prod_{l=1}^r (f_2(x_{i_l}) - f_1(x_{i_l})) \right) \prod_{i=1}^n dx_i
\end{aligned}$$

where $\mathscr{P}(n,r)$ denotes the set of all permutations of r items from $\{1,\ldots,n\}$. Now, integrating the first term in the integrand gives $P_{f_1}(S_n > b)$. Integrating the second term gives

$$\begin{aligned}
& \sum_{i=1}^n \int \left(\int \cdots \int_{\substack{\sum_{l=1,\ldots,n, l\neq i} x_l > b - x_i}} \prod_{\substack{j=1,\ldots,n \\ j\neq i}} f_1(x_j)dx_j \right) (f_2(x_i) - f_1(x_i))dx_i \\
= & \; n \int P_{f_1}(S_{n-1} > b - x)(f_2(x) - f_1(x))dx
\end{aligned}$$

by noting that x_i's are exchangeable. Note that each summand in the third term contains two or more factors of $f_2(x_{i_l}) - f_1(x_{i_l})$, and the integral of the term is absolutely bounded by

$$C \int \int |f_2(x) - f_1(x)||f_2(y) - f_1(y)|dxdy$$

for some $C > 0$. Combining the above gives (6). □

The function $\zeta(\cdot; f_1)$ can be interpreted as the first order gradient of $P_f(S_n > b)$ at f_1. When f_1 and f_2 are very close in terms of the L_1-distance, the second order term in (6) becomes relatively negligible. The function $\zeta(\cdot; f_1)$ is essentially the Gateaux derivative of $Z(f_1)$ and is related to the so-called influence function (Huber 2011) in robust statistics, which is an important tool in measuring the infinitesimal effect on a given statistics due to changes in the distribution of data.

Therefore, from (6), given a density f_1, the first order approximation of $P_{f_2}(S_n > b)$ centered at f_1 is given by

$$P_{f_2}(S_n > b) \approx P_{f_1}(S_n > b) + E_{f_2}[\zeta(X; f_1)] - E_{f_1}[\zeta(X; f_1)].$$

This suggests that at each iteration k in our optimization procedure, given f_k, one would solve $\max_{f \in \mathscr{A}} E_f[\zeta(X; f_k)]$ as the subprogram. When $\zeta(X; f_k)$ is not computable exactly, an estimated version $\widehat{\zeta(X; f_k)}$ will be substituted.

5 WORST-CASE CHARACTERIZATION FOR LINEAR OBJECTIVES

We discuss how to solve $\max_{f \in \mathscr{A}} E_f[\zeta(X)]$ for a given function $\zeta(\cdot)$. This borrows results from Lam and Mottet (2015). To formulate the optimization more explicitly, let us pin down the necessary information from the known function g. We define:

1. $\eta = g(a)$ as the value of g at a.
2. $-\nu = g'_-(a)$ as the left derivative of g at a.
3. $\beta = \int_a^\infty g(x)dx$ as the tail distribution at a, which is equal to $1 - \int_{-\infty}^a g(x)dx$.

We assume that η, ν, β are all positive. Then the formulation $\max_{f \in \mathscr{A}} E_f[\zeta(X)]$ can be written as

$$\begin{array}{ll} \max_f & \int_{-\infty}^a \zeta(x)g(x)dx + \int_a^\infty \zeta(x)f(x)dx \\ \text{subject to} & \int_a^\infty f(x)dx = \beta \\ & f(a) = \eta \\ & f'_+(a) \geq -\nu \\ & f \text{ convex for } x \geq a \\ & f(x) \geq 0 \text{ for } x \geq a. \end{array} \tag{7}$$

In this optimization, the first constraint makes sure that f is a valid probability density by equating the tail probability at a to β. The second constraint states that f has to be a continuous extrapolation from g, since f is convex. The third constraint specifies that the right derivative of f at a needs to be at least $-\nu$, because of the convexity assumption again. The whole set of constraints therefore ensures that f is a valid density that is a convex extrapolation from g.

The following result is adopted from Lam and Mottet (2015):

Theorem 1 Consider the optimization $\max_{f \in \mathscr{A}} E_f[\zeta(X)]$. Suppose that $\zeta(\cdot)$ is bounded, non-negative and non-decreasing, and $\eta, \nu, \beta > 0$. If $\eta^2 > 2\beta\nu$, there is no feasible solution. Otherwise, optimality is characterized by either one of the two scenarios:

1. There is an optimal density that is piecewise linear with at most two line segments for $x > a$. The first line segment has slope $-\nu$ and the second line segment hits the x-axis. (The first line segment can be degenerate, i.e. zero length.)
2. There is no optimal density. Instead, there is a sequence of densities $f^{(s)}, s = 1, 2, \ldots$ whose objective value converges to the optimum. Each of these $f^{(s)}$ has at most three line segments. The first line segment has slope $-\nu$ and the third line segment hits the x-axis. The third line segment becomes closer and more parallel to the x-axis as $s \to \infty$. (The first and the second line segments can be degenerate.)

Hence, solving (7) reduces to finding the kinks of piecewise linear functions. Algorithm 1 depicted below can classify between the above two scenarios and also solve for the optimal density or sequence of densities explicitly.

Let us explain the output of Algorithm 1. The output $f(x)$ encodes either an optimal density, in the first case of Theorem 1, or the pointwise limit of a sequence of densities $f^{(s)}$ that converges to optimality, in the second case of Theorem 1. We emphasize the fact that in the second case, the algorithm does not return the sequence $f^{(s)}$ but only its pointwise limit $f(x)$. Note that the limit density $f(x)$ is not a valid density itself, in the sense that it does not integrate to 1. The quantity q^* encodes the amount of probability mass that has "escaped" to infinity.

The variable x_1 represents the location of the first kink in the optimal piecewise linear density (or limit density). The function $W(x_1)$ is introduced to find this first kink and, in the second case of Theorem 1, an additional function $V(x_1,\rho)$ is needed. When $\tilde{x}_1 = \operatorname{argmax}_{x_1\in[0,\mu-\varepsilon]} W(x_1) \neq \mu-\varepsilon$, we have $q^*=0$, which signifies that the optimality characterization falls into the first case in Theorem 1. On the other hand, if $\tilde{x}_1 = \mu-\varepsilon$, then $q^*>0$ which brings to the second case. The tolerance parameter ε is introduced to detect whether the maximum of $W(x_1)$ or $V(x_1,\rho)$ occurs at $x_1=\mu$. Note that the definitions of $W(x_1)$ and $V(x_1,\rho)$ both have singularities at $x_1=\mu$ (it can be checked that the values of $W(x_1)$ and $V(x_1,\rho)$ remain bounded as $x_1\to\mu$). Introducing ε and checking $\tilde{x}_1=\mu-\varepsilon$ is an implementable way to detect that the solution is at the singularity μ.

In the last scenario in the algorithm, when $x_1^*=\mu$ and $\rho^*=\mu^2$, we define $p_1^*=1$ and $x_2^*=x_1^*$, or in other words output $f(x)=\eta-\nu(x-a)$ for $a\le x\le x_1^*+a$ and 0 for $x\ge x_1^*+a$.

Algorithm 1 Procedure for solving (7)

Input: η, ν, β, $\zeta(x)$, and a small "tolerance" parameter $\varepsilon > 0$

Definition: Let

$$
\begin{aligned}
H(x) &= \int_0^x \int_0^u \zeta(v+a)dvdu \\
\lambda &= \limsup_{x\to\infty} \frac{H(x)}{x^2} \\
W(x_1) &= \frac{\sigma-\mu^2}{\sigma-2\mu x_1+{x_1}^2} H(x_1) + \frac{\mu-x_1^2}{\sigma-2\mu x_1+x_1^2} H\left(\frac{\sigma-\mu x_1}{\mu-x_1}\right) \\
V(x_1,\rho) &= \frac{\rho-\mu^2}{\rho-2\mu x_1+x_1^2}\left(H(x_1)-\lambda x_1^2\right) + \frac{\mu-x_1^2}{\rho-2\mu x_1+x_1^2}\left(H\left(\frac{\rho-\mu x_1}{\mu-x_1}\right) - \lambda\left(\frac{\rho-\mu x_1}{\mu-x_1}\right)^2\right) + \lambda\sigma
\end{aligned}
$$

Initialization:

$\mu \leftarrow \frac{\eta}{\nu}$

$\sigma \leftarrow \frac{2\beta}{\nu}$

$q^* \leftarrow 0$ ▷ *By default, the escaping mass is null.*

Procedure:

if $\sigma < \mu^2$ **then**

  STOP. There is no feasible solution.

else if $\sigma = \mu^2$ **then**

  $\tilde{x}_1 \leftarrow \mu$

  $\rho^* \leftarrow \mu^2$

} *Exclusion of trivial scenarios.*

else

  $\tilde{x}_1 \leftarrow \operatorname{argmax}_{x_1\in[0,\mu-\varepsilon]} W(x_1)$

  if $\tilde{x}_1 \neq \mu-\varepsilon$ **then**

    $\rho^* \leftarrow \sigma$

  } *Case 1 : Light tail.*

  else

    $(\tilde{x}_1,\rho^*) \leftarrow \operatorname{argmax}_{x_1\in[0,\mu-\varepsilon],\rho\in[\mu^2,\sigma]} V(x_1,\rho)$

    if $\tilde{x}_1 = \mu-\varepsilon$ **then**

      $\tilde{x}_1 \leftarrow \mu$

      $\rho^* \leftarrow \mu^2$

    } *Case 2 : Heavy tail.*

    end if

    $q^* \leftarrow \frac{\nu}{2}(\sigma-\rho^*)$

  end if

} *Treatment of non-trivial scenarios.*

$x_1^* \leftarrow \tilde{x}_1$

$p_1^* \leftarrow \frac{\rho^*-\mu^2}{\rho^*-2\mu x_1^*+{x_1^*}^2}$

$x_2^* \leftarrow \frac{\rho^*-\mu x_1^*}{\mu-x_1^*}$

$$
f(x) \leftarrow \begin{cases} \eta-\nu(x-a) & \text{for } a \le x \le x_1^*+a \\ \eta-\nu x_1^*-\nu(1-p_1^*)(x-a-x_1^*) & \text{for } x_1^*+a \le x \le x_2^*+a \\ 0 & \text{for } x \ge x_2^*+a \end{cases}
$$

end if

return $(f(x),q^*)$

6 AUGMENTED PROBABILITY DENSITIES

6.1 The Calculus of Augmented Probability Densities

The type of outputs in Algorithm 1 forms a natural space for running our FW algorithm, and here we shall describe it in more detail. We define an "augmented" probability density as $(f(x),q)$, where $f(x) \geq 0$ for $-\infty < x < \infty$, $\int_{-\infty}^{\infty} f(x)dx \leq 1$, and $q = 1 - \int_{-\infty}^{\infty} f(x)dx$.

We define the expectation of a bounded function $h(x)$, with limit $h(\infty) = \lim_{x\to\infty} h(x)$, under an augmented density $(f(x),q)$ as

$$E_{f,q}[h(X)] = \int_{-\infty}^{\infty} h(x)f(x)dx + h(\infty)q. \tag{8}$$

More complicated functionals of an augmented density can be defined by viewing q as an "escaping" probability mass to positive infinity. For instance, for X_1, X_2 i.i.d. with augmented density $(f(x),q)$, we would define

$$P_{f,q}(X_1 + X_2 > b) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x_1 + x_2 > b)f(x_1)f(x_2)dx_1dx_2 + (1 - (1-q)^2).$$

To interpret the above, one can think that if say a particular realization of X_1 has "escaped", then X_1 must be larger than b. The first term above corresponds to the part that both X_1 and X_2 do not escape. The second part corresponds to the scenario that at least one of them escape.

To simulate quantities dictated by an augmented density $(f(x),q)$, one can use a mixture of $\tilde{f}(x) = f(x)/(1-q)$ and the escaping mass q, by noting that $\tilde{f}(x)$ is a valid density. For instance, for X_1, X_2 i.i.d. with augmented density $(f(x),q)$,

$$\begin{aligned} P_{f,q}(X_1 + X_2 > b) &= (1-q)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x_1 + x_2 > b)\frac{f(x_1)}{1-q}\frac{f(x_2)}{1-q}dx_1dx_2 + (1 - (1-q)^2) \\ &= (1-q)^2 P_{\tilde{f}}(X_1 + X_2 > b) + (1 - (1-q)^2) \end{aligned} \tag{9}$$

where $P_{\tilde{f}}(\cdot)$ is the probability measure generated by i.i.d. X_1 and X_2 each under the density $\tilde{f}$, and $(1-(1-q)^2)$ above can be evaluated exactly.

For another example, say $P_{f,q}(X_1 + X_2 \in [b,c])$ for X_1, X_2 i.i.d. with augmented density $(f(x),q)$, we can express as

$$P_{f,q}(X_1 + X_2 \in [b,c]) = (1-q)^2 P_{\tilde{f}}(x_1 + x_2 > b).$$

The term corresponding to the escaping mass disappears now because if say X_1 escapes, then the event $X_1 + X_2 \in [b,c]$ cannot happen.

6.2 Consistency between Optimal Density Sequence and Augmented Density

One key observation when building the above rules for augmented densities is that they are consistent with the limiting behavior of the optimal sequence of densities in the second case of Theorem 1. It can be shown that for an optimal density sequence for problem (7) given by $f^{(s)}$ that has pointwise limit f,

$$E_{f,q}[h(X)] = \lim_{s\to\infty} E_{f^{(s)}}[h(X)]$$

where $E_{f,q}[h(X)]$ is defined by rule (8). Moreover, consistency is also preserved for the tail probability of i.i.d. sum, namely

$$P_{f,q}(S_n > b) = \lim_{s\to\infty} P_{f^{(s)}}(S_n > b) = (1-q)^n P_{\tilde{f}}(S_n > b) + (1 - (1-q)^n)$$

where $P_{f,q}$ is defined by rule (9) and $\tilde{f}(x) = f(x)/(1-q)$. The proof of these results will be shown elsewhere.

7 ITERATIVE SCHEME VIA FRANK-WOLFE STOCHASTIC APPROXIMATION

Our FW algorithm will operate in the augmented density space. Since the linearization at each iteration has to be simulated, it can be viewed as a constrained stochastic approximation (SA) method. It is stated as:

Algorithm 2 Iterative procedure for solving (2)

Input: the function $g(x), x \leq a$, the level b, step size ε_k, sample size m_k for each step k.
Initialization: an augmented density $(f_1(x), q_1)$ where $f_1(x)$ meets the constraints of problem (7). For instance, one can take $(f_1(x), q_1)$ to be the augmented density representing $\text{argmax}_{f \in \mathscr{A}} P_f(X > b)$, i.e. the output from Algorithm Algorithm 1 by putting $\zeta(x) = I(x > b)$ where $I(\cdot)$ denotes the indicator function. (Other initializations will also work.)
Procedure: For each iteration $k = 1, 2, \ldots$, given an augmented density $(f_k(x), q_k)$:

1. Compute

$$\hat{\zeta}_k(x) = \frac{1}{m_k} \sum_{i=1}^{m_k} I(S_{n-1} > b - x)$$

using m_k sample paths of S_{n-1}, where $S_{n-1} = X_1 + \cdots + X_{n-1}$ for i.i.d. X_i's generated under $\tilde{f}_k(x) = f_k(x)/(1-q_k)$.
2. Run Algorithm 1 with input function $\hat{\zeta}_k(x)$ to get an output $(r_k(x), u_k)$.
3. Update $(f_{k+1}(x), q_{k+1}) = (1-\varepsilon_k)(f_k(x), q_k) + \varepsilon_k(r_k(x), u_k)$.

We explain some details related to the subprogram step. By Lemma 1, one should post the subprogram at iteration k as $\max_{f \in \mathscr{A}} E_f[nP_{f_k,q_k}(S_{n-1} > b - X)]$, where $(f_k(x), q_k)$ is the current augmented density. Then by the rules in Section 6.1 we have

$$P_{f_k,q_k}(S_{n-1} > b - x) = (1-q_k)^{n-1} P_{\tilde{f}_k}(S_{n-1} > b - x) + (1 - (1-q_k)^{n-1}).$$

Hence optimizing $E_f[P_{f_k,q_k}(S_{n-1} > b - X)]$ is equivalent to optimizing $E_f[P_{\tilde{f}_k}(S_{n-1} > b - x)]$. The quantity $\hat{\zeta}_k(x)$ is an unbiased estimator for $P_{\tilde{f}_k}(S_{n-1} > b - x)$.

Note that the updating step $(f_{k+1}(x), q_{k+1}) = (1-\varepsilon_k)(f_k(x), q_k) + \varepsilon_k(r_k(x), u_k)$ represents a mixture of $(f_k(x), q_k)$ and $(r_k(x), u_k)$, which is defined as the mixture over both the density part and the escaping mass part. In other words, we have $f_{k+1}(x) = (1-\varepsilon_k) f_k(x) + \varepsilon_k r_k(x)$ and $q_{k+1} = (1-\varepsilon_k) q_k + \varepsilon_k u_k$.

8 NUMERICAL EXAMPLE

Consider the setting where f is known to be an exponential distribution with rate λ, up to a. We apply Algorithm 2 to estimate a local optimum for $\max_{f \in \mathscr{A}} P_f(S_n > b)$. We set $n = 8$, $b = 10$, and $\lambda = 1$. We vary a from the 70-th percentile of the exponential distribution to the 99-th percentile. For each a, we apply our Algorithm to find the locally optimal augmented density, and we simulate $P(S_n > b)$ using that to get an estimate of the maximum objective value.

Note that even though the gradient estimator $\hat{\zeta}_k(x)$ at each iteration is unbiased, the estimated best feasible direction obtained through solving the optimization subprogram is in general biased. Consequently, in our implementation we use a growing sample size $m_k = 10k^{1.1}$ along the iterations. We use a standard SA step size specification $\varepsilon_k = 1/k$. Some theoretical guarantees regarding this specification for FWSA, though in a separate context, are reported in Ghosh and Lam (2015).

Since g is taken to be $Exp(1)$, we know that the actual distribution of S_n is a Gamma distribution with shape and rate parameters equal to n and 1 respectively. The actual value of $P(S_n > b)$ is therefore the tail distribution of the Gamma distribution evaluated at the point b. This provides a benchmark for measuring the level of conservativeness of the output from Algorithm 2.

Figure 1 shows the estimated value of $P_{f_{100}}(S_n > b)$, i.e. we ran 100 iterations, for different values of the threshold a. In terms of convergence, we found that in this example Algorithm 2 almost reached equilibrium in one iteration, and so 100 iterations was likely more than enough. On the other hand, we also tried the same set of experiments for the corresponding minimization problem, and found that the algorithm converged in around 20 iterations. In Figure 1, it can be seen that, when a is at the 70-th percentile, the estimate of $P(S_n > b)$ of about 0.7 is very conservative, compared to the true value of around 0.2 dictated by the Gamma distribution. Nevertheless, the estimate gets progressively less conservative as a gets larger, until it drops to almost the true value at the 99-th percentile.

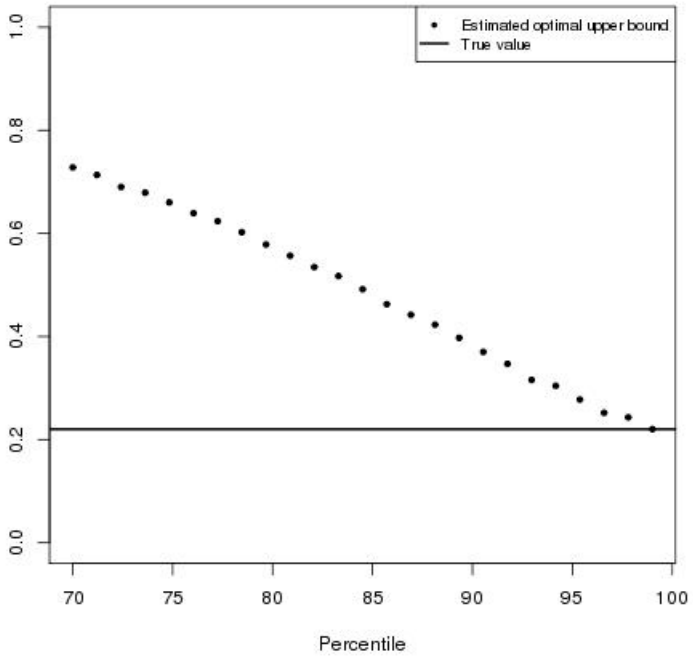

Figure 1: Estimated optimal upper bound of $P_{f_{100}}(S_8 > 10)$ for $X_i \sim Exp(1)$ known up to different percentiles.

Lastly, we provide some discussion on how to simulate $\tilde{f}_k$ at each iteration, which is used to generate copies of S_{n-1} and subsequently $\hat{\zeta}_k(x)$ in Algorithm 2. To simulate a copy from $\tilde{f}_k$, we can use the method of inverse distribution function. First, since f_k is a mixture of augmented densities, we start with generating one of the components with the mixture probabilities. This component will have density $\lambda e^{-\lambda x}$ up to a, and then followed by a straight line with some slope $-s_1$ up to x_1, and slope $-s_2$ up to x_2 which touches the x-axis. To simulate from this component conditional on being not escaping, we can generate a uniform

random variable U over the interval $[0, 1-q_k]$. Then, letting $F(a) = 1-e^{-\lambda x}$ and $f(a) = \lambda e^{-\lambda a}$, we output

$$\begin{cases} -\frac{1}{\lambda}\log(1-U) \\ \qquad\qquad\qquad \text{for } \; U \le F(a) \\ a + \frac{f(a)+\sqrt{f(a)^2-2s_1(U-F(a))}}{s_1} \\ \qquad\qquad\qquad \text{for } \; F(a) \le U \le F(a)+f(a)(x_1-a)-\frac{s_1(x_1-a)^2}{2} \\ x_1 + \frac{f(a)-s_1(x_1-a)+\sqrt{(f(a)-s_1(x_1-a))^2-2s_2(U-F(a)-f(a)(x_1-a)+\frac{s_1(x_1-a)^2}{2})}}{s_2} \\ \qquad\qquad\qquad \text{for } \; F(a)+f(a)(x_1-a)-\frac{s_1(x_1-a)^2}{2} \le U \le x_2. \end{cases}$$

This can be seen by solving for the inverse at each segment (the exponential, and the two quadratic segments) in the distribution function. For the first quadratic segment, one needs to solve, for a given U,

$$F(a)+f(a)(x-a)-\frac{s_1(x-a)^2}{2} = U$$

in x, and for the second quadratic segment, one solves

$$F(a)+f(a)(x_1-a)-\frac{s_1(x_1-a)^2}{2}+(f(a)-s_1(x_1-a))(x-x_1)-\frac{s_2(x-x_1)^2}{2} = U.$$

9 DISCUSSION AND FUTURE WORK

This paper demonstrates an iterative scheme, based on a stochastic version of the FW method, to find worst-case estimates of the tail probability of i.i.d. sum assuming the distribution of the summands is known only up to a specific threshold. We have described how to handle the infinite-dimensional nature of the problem, and also the possibility of non-existence of an optimal solution in any stepwise subprogram by introducing the notion of augmented probability density. Several future investigation directions are in line. The first to obtain some theoretical convergence guarantees of the proposed algorithm. Second, for a high threshold b, naive Monte Carlo for estimating $\hat{\zeta}_k(x)$ in each iteration, like what we have used, will be too slow. The use of importance sampling will be investigated. Third, we also plan to generalize the result to other types of objectives in addition to the tail probability of i.i.d. sum. Lastly, all results in this paper assume complete information on the non-tail part of the distribution, whereas in most cases it is partially known through data. In the future, we plan to extend our analysis to joint uncertainty on both the tail and the non-tail regions.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the National Science Foundation under grants CMMI-1400391 and CMMI-1436247.

REFERENCES

Ben-Tal, A., D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. 2013. "Robust Solutions of Optimization Problems Affected by Uncertain Probabilities". *Management Science* 59 (2): 341–357.

Bertsekas, D. P. 1999. *Nonlinear Programming*. Athena scientific Belmont.

Birge, J. R., and R. J.-B. Wets. 1987. "Computing Bounds for Stochastic Programming Problems by Means of a Generalized Moment Problem". *Mathematics of Operations Research* 12 (1): 149–162.

Delage, E., and Y. Ye. 2010. "Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems". *Operations research* 58 (3): 595–612.

Ghosh, S., and H. Lam. 2015. "Computing Worst-Case Input Models in Stochastic Simulation". *submitted, available at: http://www-personal.umich.edu/ khlam/files/robustSAOR3.pdf*.

Goeva, A., H. Lam, and B. Zhang. 2014. "Reconstructing Input Models via Simulation Optimization". In *Proceedings of the 2014 Winter Simulation Conference*, 698–709. IEEE Press.

Huber, P. J. 2011. *Robust Statistics*. Springer.

Lam, H., and C. Mottet. 2015. "Tail Analysis Without Tail Information: A Worst-Case Perspective". *submitted, available at: http://www-personal.umich.edu/ khlam/files/tailOR7.pdf*.

Popescu, I. 2005. "A Semidefinite Programming Approach to Optimal-Moment Bounds for Convex Classes of Distributions". *Mathematics of Operations Research* 30 (3): 632–657.

Smith, J. E. 1995. "Generalized Chebychev Inequalities: Theory and Applications in Decision Analysis". *Operations Research* 43 (5): 807–825.

AUTHOR BIOGRAPHIES

HENRY LAM is an Assistant Professor in the Department of Industrial and Operations Engineering at the University of Michigan, Ann Arbor. He graduated from Harvard University with a Ph.D. degree in statistics in 2011, and has been an Assistant Professor in the Department of Mathematics and Statistics at Boston University until 2014. His research focuses on stochastic simulation, rare-event analysis, and simulation optimization, with application interests in service systems and risk management. His email address is khlam@umich.edu.

CLEMENTINE MOTTET is a Ph.D. student in the Department of Mathematics and Statistics at Boston University. She joined their PhD program in 2013 after graduating from the the National Institute of Applied Sciences of Toulouse with a Master's Degree in Applied Mathematics. She is currently conducting research under the supervision of Professor Lam. Their collaborative research has been focusing on robust estimation of tail quantities under limited information, which is relevant in fields related to risk management. She can be reached at cmottet@bu.edu.