Scenario generation for stochastic optimization problems via the sparse grid method [*]

Michael Chen[†] Sanjay Mehrotra[‡] Dávid Papp[§]

March 7, 2012

Abstract

We study the use of sparse grids methods for the scenario generation (or discretization) problem in stochastic programming problems where the uncertainty is modeled using a continuous multivariate distribution. We show that, under a regularity assumption on the random function, the sequence of optimal solutions of the sparse grid approximations, as the number of scenarios increases, converges to the true optimal solutions. The rate of convergence is also established. We consider the use of quadrature formulas tailored to the stochastic programs where the uncertainty can be described via a linear transformation of a product of univariate distributions, such as joint normal distributions. We numerically compare the performance of the sparse grid method using different quadrature rules with quasi-Monte Carlo (QMC) methods and Monte Carlo (MC) scenario generation, using a series of utility maximization problems with up to 160 random variables. The results show that the sparse grid method is very efficient if the integrand is sufficiently smooth. In such problems the sparse grid scenario generation method is found to need several orders of magnitude fewer scenarios than MC and QMC scenario generation to achieve the same accuracy. The method is potentially scalable to problem with thousands of random variables.

Keywords: Scenario generation, Stochastic optimization, Discretization, Sparse grid

[*] DOE DE-FG02-10ER26037, SC0005102; DOE-SP0011568; and ONR N00014210051. An earlier technical report version of this paper was distributed under the title "Scenario Generation for Stochastic Problems via the Sparse Grid Method".

[†] Department of Mathematics and Statistics, York University, Toronto, Canada, chensy@mathstat.yorku.ca

[‡] Department of IE/MS, Northwestern University, Evanston, IL 60208, mehrotra@iems.northwestern.edu

[§] Department of IE/MS, Northwestern University, Evanston, IL 60208, dpapp@iems.northwestern.edu

1 Introduction

Stochastic optimization is a fundamental tool in decision making, with a wide variety of applications (see, e.g., Wallace and Ziemba [34]). A general formulation of this problem is

$$\min_{x\in\mathcal{C}} \int_{\Xi} f(\xi,x)P(d\xi), \tag{1}$$

where the n-dimensional random parameters ξ are defined on a probability space $(\Xi,\mathcal{F},P)$, $\mathcal{F}$ is the Borel σ-field on Ξ and P represents the given probability measure modeling the uncertainty in the problem, and $\mathcal{C}\subseteq\mathbb{R}^{\bar{n}}$ is a given (deterministic) set of possible decisions.

In practice, the continuum of possible outcomes corresponding to P is often approximated by a finite number of scenarios ξ^k ($k=1,\dots,K$) with positive probabilities w^k, which amounts to the approximation of the integral in (1) by a finite sum:

$$\min_{x\in\mathcal{C}} \int_{\Xi} f(\xi,x)P(d\xi) \approx \min_{x\in\mathcal{C}} \sum_{k=1}^{K} w^k f(\xi^k,x). \tag{2}$$

The efficient generation of scenarios that achieve good approximation in (2) is, thus, a central problem in stochastic programming.

Several methods have been proposed for the solution of this problem. The popular *Monte Carlo (MC) method* uses pseudo-random numbers (vectors) as scenarios and uniform weights $w^1=\cdots=w^K=1/K$. Pennanen and Koivu [27] use *quasi-Monte Carlo (QMC) methods*, that is, low-discrepancy sequences, the Faure sequence, Sobol sequence, or the Niederreiter sequence, as scenarios along with uniform weights. *Scenario reduction methods*, e.g [4, 14] generate a large number of scenarios, and select a small, "good", subset of them, using different heuristics and data mining techniques, such as clustering and importance sampling. Dempster and Thompson [6] use a heuristic based on the value of perfect information, with parallel architectures in mind. *Optimal discretization* approaches [9, 29] choose the scenarios by minimizing a probability metric. Casey and Sen [3] apply linear programming sensitivity analysis to guide scenario generation.

King and Wets [16] and Donohue [7] studied epi-convergence of the Monte-Carlo method. Epi-convergence of the QMC methods is established in Pennanen and Koivu [28], and Pennanen [26]. Since establishing the rate of convergence is difficult (for example, results for QMC methods are unknown), different scenario generation methods are usually compared to each other numerically. Pennanen and Koivu [28] tested QMC methods on the Markowitz model. The Markowitz model has an alternate closed form expression of the integral, which allows one to test the quality of the approximation by comparing the objective value from the approximated model with the true optimal objective value. If the true optimal value is unknown, a statistical upper and lower bound obtained from Monte Carlo sampling (see, e.g, [22]) may be used to compare scenario generation methods. Kaut and Wallace [15] give further guidelines for evaluating scenario generation methods.

The main contributions are this paper are twofold. Two variants of a sparse grid scenario generation method are proposed for the solution of stochastic optimization problems.

After establishing their convergence and rate of convergence (Theorems 4 and 5), we numerically compare them to MC and QMC methods on a variety of problems which differ in their dimensionality, objective, and underlying distribution. The results show that the sparse grid method compares favorably with the Monte Carlo and QMC methods for smooth integrands, especially when the uncertainty is expressible using distributions that are linear transformations of a product of univariate distributions (such as the multinormal distribution). In such problems, the sparse grid scenario generation method is found to need several orders of magnitude fewer scenarios than MC and QMC scenario generation to achieve the same accuracy.

The paper is organized as follows. In Section 2 we review the Smoljak's sparse grid approximation method and its application for scenario generation. In Section 3 we adapt this method for stochastic optimization. The general method of is presented in Section 3.1, whereas the variant presented Section 3.2 is designed for problems that can be transformed linearly to a problem whose underlying probability distribution is a product of univariate distributions. In Section 4 we show that the sparse grid method gives an exact representation of (1) if the integrand function belongs to a certain polynomial space. Here we also show that the method is uniformly convergent and that the rate of convergence of sparse grid scenario generation is the same as the rate of convergence of sparse grid numerical integration methods. Numerical results follow in Section 5. We give results both on stochastic optimization problems which have been used for testing in the literature, and on considerably larger new simulated examples. The motivating application in these examples is portfolio optimization with various utility functions as objectives. The results show numerically that the sparse grid method compares favorably with the Monte Carlo and QMC methods for smooth integrands, and scales very well. Additional remarks are made in Section 6.

2 Sparse Grid Scenario Generation in Numerical Integration

2.1 Quadrature rules for numerical integration

The sparse grid scenario generation method uses a univariate *quadrature rule* as a basic ingredient. A quadrature rule gives, for every $\nu \in \mathbb{N}$, a set of points (nodes) $\omega_k \in \mathbb{R}$ and corresponding weights $w_k \in \mathbb{R}$, $k = 1, \dots, L(\nu)$, used to approximate a one-dimensional integral:

$$\int_\Omega f(\omega)\rho(\omega)d\omega \approx \sum_{k=1}^{L(\nu)} w_k f(\omega_k), \tag{3}$$

where Ω is either a closed bounded interval (without loss of generality, $[0, 1]$) or the real line $\mathbb{R}$, and $\rho\colon \Omega \mapsto \mathbb{R}_+$ is a given nonnegative, measurable *weight function*. For the stochastic optimization problems of our concern it is sufficient to consider probability density functions of continuous random variables that are supported on Ω, and which have finite moments of every order $m \in \mathbb{N}$. For a given univariate quadrature rule the number of scenarios is specified by the function $L(\nu) : \mathbb{N} \to \mathbb{N}$, where ν is called the *resolution* of the formula. The univariate quadrature rules differ in $L(\cdot)$, and in the nodes and weights they generate for a fixed ν.

Examples of univariate quadrature rules for $\Omega = [0, 1]$ with the constant weight function $\rho = 1$ include the Newton–Cotes (midpoint, rectangle, and trapezoidal) rules, the Gauss–Legendre rule, the Clenshaw–Curtis rule, and the Gauss–Kronrod–Patterson (or GKP) rule; see, for example, Davis and Rabinowitz [5] or Neumaier [19] for the definitions of these rules.

Examples of quadrature rules for other domains and commonly used weight functions, including the case when $\Omega = \mathbb{R}$ and $\rho(\omega) = \exp(-\omega^2)$, can be found in Krylov's monograph [17]. We mention two important families of such rules. *Gaussian quadrature* rules [17, Chap. 7] are the generalization of Gauss–Legendre rules, and can be defined for every domain and weight function; they satisfy $L(\omega) = \omega$. We introduce the second family, *Patterson-type rules* in Section 3.2; these rules have an exponentially increasing function L.

An important feature of quadrature rules is their *degree of polynomial exactness* (sometimes called degree of precision), which is the largest integer D_ν for which the approximation (3) with resolution ν is exact for every polynomial of degree up to D_ν. For example, for the Gaussian rules we have $D_\nu = 2\nu - 1$, whereas the GKP rule satisfies $D_1 = 1$ and $D_\nu = 3 \cdot 2^{\nu-1} - 1$, for $\nu \geq 2$. [13]. High degree of polynomial exactness translates to a good approximation of the integrals of functions approximable by polynomial functions. For r-times weakly differentiable functions f (see also (7) below) the error in the approximation (3) can be estimated as:

$$\left| \int_\Omega f(\omega)\rho(\omega)d\omega - \sum_{k=1}^{L(\nu)} w_k f(\omega_k) \right| = O\left(K^{-r}\right),$$

where $K = L(\nu)$ is the number of nodes.

Let $\Omega_\nu = \{\omega_1, \dots, \omega_{L(\nu)}\}$ be the nodes and $w_{\omega_\nu} = \{w_1, \dots, w_{L(\nu)}\}$ be the corresponding weights. If $\Omega_\nu \subset \Omega_{\nu+1}$, then the univariate quadrature rule is called *nested*. The nodes of Gaussian quadrature rules are not nested. Examples of nested univariate quadrature rules are the iterated trapezoidal, the nested Clenshaw–Curtis, and the Gauss–Knonrod–Patterson (GKP) formulas [13], as well as the Patterson-type rules of Section 3.2. In general, nested quadrature rules are preferred in the sparse grid scenario generation, as they are known to give better approximations to the integrals than sparse grid formulas with non-nested rules, such as the Gaussian.

2.2 Smolyak's Sparse Grid Construction

The sparse grid algorithm by Smolyak [32] utilizes any of the above univariate quadrature rules and uses a complexity parameter to limit the number of grid point it generates. Sparse grids have significantly fewer nodes than the exponential number of nodes of the product grid obtained by taking direct products of univariate quadrature formulas. We now briefly describe Smolyak's construction of multivariate quadrature rule and summarize the main results on this topic. See Bungartz and Griebel [1] for a review on sparse grid methods (with a focus on differential equations).

Let $\rho(\omega_1, \ldots, \omega_n) = \rho_1(\omega_1) \cdots \rho_n(\omega_n)$ be a given n-variate weight function, $\nu_1, \cdots, \nu_n$ represent the sparse grid resolution along dimensions $1, \ldots, n$, and ν be a shorthand for $(\nu_1, \cdots, \nu_n)$. Let $\Omega_{\nu_i} = \{\omega_1, \ldots, \omega_{L(\nu_i)}\}$ be the nodes of the univariate quadrature rule for the weight function ρ_i with resolution ν_i, and let $w_1, \ldots, w_{L(\nu_i)}$ be the corresponding weights. For a given positive integer q, Smolyak's sparse grid formula uses the nodes in

$$\mathcal{G}(q, n) = \bigcup_{\|\nu\|_1 \leq q+n-1} (\Omega_{\nu_1} \times \cdots \times \Omega_{\nu_n}), \tag{4}$$

and approximates the multivariate integral

$$\int_{\Omega^n} f(\omega)\rho(\omega) d\omega \tag{5}$$

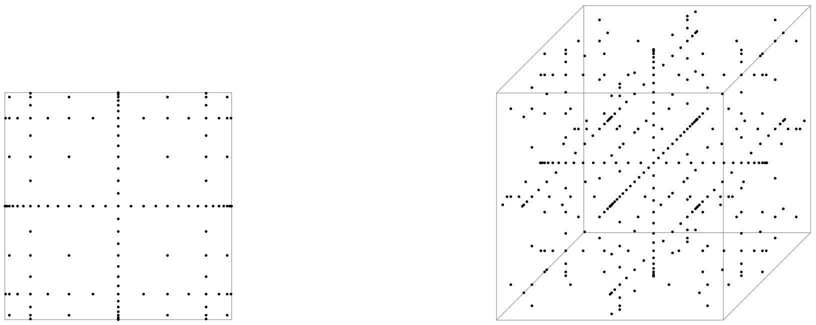
by the finite sum

$$\sum_{k=1}^{K} w^k f(\omega^k) := \sum_{q \leq \|\nu\|_1 \leq q+n-1} (-1)^{q+n-\|\nu\|_1-1} \begin{pmatrix} n-1 \\ \|\nu\|_1 - q \end{pmatrix} \sum_{k_1=1}^{L_{\nu_1}} \cdots \sum_{k_n=1}^{L_{\nu_n}} w_{k_1} \cdots w_{k_n} f(\omega_{k_1}, \ldots, \omega_{k_n}). \tag{6}$$

We now present results on the error estimates for the sparse grid method in approximating a multi-variate integral. Theorem 1 shows that Smolyak's multivariate quadrature preserves the underlying univariate quadrature rule polynomial exactness property. For example, if $n = 2, q = 2$, then using the Gaussian quadrature rule the approximation is exact for polynomials that are a linear combination of $x, x^2, x^3, y, y^2, y^3, xy, x^2y, xy^2, x^3y, xy^3$, and a constant. Hence, the approximation is exact for all polynomials of degree 3, and monomials x^3y, xy^3 of degree 4. In general, for any n and $q = 2$ the approximation is exact for polynomials of degree 3. A general result on polynomial exactness is restated in the following theorem.

Theorem 1. *(Gerstner and Griebel [13], Novak and Ritter [23]) Let $q \in \mathbb{N}$, and the sparse grid approximation of (5) be given as in (6). Let D_{ν_i} be the degree of polynomial exactness of a univariate quadrature rule with resolution ν_i. Let $\mathcal{P}_{D_{\nu_i}} \otimes \mathcal{P}_{D_{\nu_j}}$ represent the space of polynomials generated by linear combination of products of monomials $p_{k_i}(x_i)$ with $p_{k_j}(x_j)$, where the monomial degree $k_i \leq D_{\nu_i}$ and $k_j \leq D_{\nu_j}$. Then the value of (6) is equal to the integral (5) for every polynomial $f \in \mathbb{P}^n_q := \sum_{\|\nu\|_1 \leq q+n-1} \left\{\mathcal{P}_{D_{\nu_1}} \otimes \cdots \otimes \mathcal{P}_{D_{\nu_n}}\right\}$.*

In Theorem 2 we restate a result providing an error bound on the approximation, when the integrand is not a polynomial. In this theorem the functional space includes functions with *weak derivatives*. Weak derivatives are defined for integrable, but not necessarily differentiable functions [10, 31]. The constant $c_{r,n}$ in Theorem 2 depends on dimension n, the order of differentiability r, and the underlying univariate quadrature rule used by the sparse grid method. Although for some cases $c_{r,n}$ is known (see Wasilkowski and Woźniakowski [35]), in general one can not expect to know $c_{r,n}$ a priori.



(a) $d = 2$

(b) $d = 3$

Figure 1: Sparse grid on unit square and unit cube for $q = 5$ with underlying GKP univariate quadrature. There are 129 scenarios in the unit square and 351 scenarios in the unit cube.

Theorem 2. *(Smolyak [32], Novak and Ritter [23], Gerstner and Griebel [13]) Consider the functional space*

$$\mathcal{W}_n^r := \left\{ f : \Omega^n \to \mathbb{R}, \left\| \frac{\partial^{s_1+s_2+\ldots+s_n} f}{\partial\omega_1^{s_1} \ldots \partial\omega_n^{s_n}} \right\|_\infty < \infty, \max(s_1, s_2, \ldots, s_n) \leq r \right\}, \tag{7}$$

where the symbol $\frac{\partial f}{\partial \omega_j}$ represents the weak-derivative of a function f w.r.t. ω_j. Assume that the chosen univariate quadrature rule satisfies $L(1) = 1$, $L(\nu) = O(2^\nu)$. For $n, r \in \mathbb{N}$, $f \in \mathcal{W}_n^r$, then K:=$|\mathcal{G}(q, n)| = O(2^q q^{n-1})$ is the cardinality of the sparse grid. Furthermore, for some $0 < c_{r,n} < \infty$ we have

$$\sup_{f \in \mathcal{W}_n^r} \left| \int_{\Omega^n} f(\omega)\rho(\omega)d\omega - \sum_{k=1}^{K} w^k f(\omega^k) \right| \leq c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} \|f\|_\infty. \tag{8}$$

From a complexity point of view the sparse grid method generates $O(2^q q^{n-1})$ grid points when $O(2^\nu)$ points are generated for a (nested) univariate quadrature rule. This is in contrast to the full grid construction where the number grows as $O(2^{qn})$, with resolution q along each dimension. Table 1 compares the full grid and sparse grid number of scenarios required to achieve a degree of polynomial exactness. It shows that even the product formula with the fewest nodes (the product of Gaussian quadrature formulas) does not scale to dimensions higher than approximately 5. In contrast, sparse grid formulas using GKP univariate quadrature rules can achieve at least moderate degree of exactness for high-dimensional problems. Figure 1 shows Smolyak's sparse grid points using GKP univariate quadrature algorithm for two and three dimensions using $q = 5$.

We emphasize that sparse grid method overcomes the "curse of dimensionality" by benefiting from the differentiability of the integrand. For a given problem of dimension n, the integration error goes to zero fast for sufficiently differentiable functions since for $r \geq 2$ in (8) the term K^{-r} dominates $(\log K)^{(n-1)(r+1)}$. In comparison, the QMC method

Dimension n,		Full grid	Sparse grid
degree d		(Gaussian)	(GKP)
$n = 5$			
	$d = 3$	32	11
	$d = 5$	243	71
	$d = 7$	1,024	351
	$d = 9$	3,125	1,471
	$d = 11$	7,776	5,503
	$d = 13$	16,807	18,943
$n = 10$			
	$d = 3$	1,024	21
	$d = 5$	59,049	241
	$d = 7$	1,048,576	2,001
	$d = 9$	9,765,625	13,441
	$d = 11$	$6.0466 \cdot 10^{7}$	77,505
$n = 20$			
	$d = 3$	1,048,576	41
	$d = 5$	$3.4868 \cdot 10^{9}$	881
	$d = 7$	$1.0995 \cdot 10^{12}$	13,201
	$d = 9$	$9.5367 \cdot 10^{13}$	154,881
$n = 50$			
	$d = 3$	$1.1259 \cdot 10^{15}$	101
	$d = 5$	$7.1790 \cdot 10^{23}$	5,201
	$d = 7$	$1.2677 \cdot 10^{30}$	182,001
$n = 200$			
	$d = 3$	$1.6070 \cdot 10^{60}$	401
	$d = 5$	$2.6561 \cdot 10^{95}$	80,801

Table 1: Number of nodes in the most efficient product grid (product of Gaussian formulas) and in the sparse grid created using GKP quadrature formulas, for various dimensions and degrees of exactness.

minimize discrepancy on the unit cube $[0,1]^n$ (Sobol [33] and Niederreiter [21, 20]). A typical convergence rate bound (see, for example Caflisch [2]) for QMC methods (Halton, Sobol, or Niederreiter sequences) is given by

$$c_n K^{-1}(\log K)^n, \tag{9}$$

where c_n is a constant. The bound in (9) does not benefit from the differentiability properties of f. We will prove in Section 3 that the sparse grid convergence results hold for stochastic optimization problems. To the best of our knowledge the rate of convergence bounds such as (9) are not yet established for stochastic optimization problems with scenario generation using QMC methods.

3 Sparse in stochastic optimization

3.1 Sparse grid scenario generation via diffeomorphism

The integration domain of the sparse grid method presented in Section 2.2 is of the form $[0,1]^n$, which gives a straightforward application to expected value computation for distributions supported on the same domain. For more general distributions we need to perform a change of variables before applying the sparse grid method. Suppose that g is a continuously differentiable diffeomorphism that generates $\xi \in \Xi$ with probability measure P from a uniform random vector $\omega \in (0,1)^n$. Then, from Folland [11, Theorem 2.47(b)]

$$\int_{\Xi} f(\xi)P(d\xi) = \int_{g((0,1)^n)} f(\xi)P(d\xi) = \int_{(0,1)^n} f(g(\omega))d\omega.$$

To approximate the integral $\int_{\Xi} f(\xi)P(d\xi)$ we proceed as follows:

1. Choose a $q \geq 1$, and generate the set $\mathcal{H}(q,n) \subset (0,1)^n$ of K scenarios and the corresponding weights by Smoljak's sparse grid construction for integration with respect to the constant weight function over $(0,1)^n$.

2. Apply (pointwise) the transformation g to $\mathcal{H}(q,n)$ to generate the stochastic optimization scenarios.

3. Use the transformed scenarios to approximate the integral $\int_{\Xi} f(\xi)P(d\xi)$.

Note that while the sparse grid formula for integrating with respect to a product measure has the degree of exactness of our choice, the same does not hold for the transformed formula for integrating over Ξ with respect to P. In other words, $f(\cdot)$ is not a polynomial when $f(g(\cdot))$ is. Only if g is linear, is the degree of exactness of the formula preserved. Hence, sparse grid formulas with a prescribed degree of exactness can be computed for every probability measure that is a linear transform of a product of probability measures on $\mathbb{R}$. An important special case is that of the multivariate normal distribution.

Example 1. *Let $\mu \in \mathbb{R}^d$ be arbitrary, and $\Sigma \in \mathbb{R}^{d\times d}$ be a positive definite matrix with spectral decomposition $\Sigma = U\Lambda U^{\mathrm{T}}$. If X is a random variable with standard multinormal distribution, then the variable $Y = \mu + U\Lambda^{1/2}X$ is jointly normally distributed with mean vector μ and covariance matrix Σ. Therefore, a sparse grid formula with degree of exactness D can be obtained from any univariate quadrature rule consisting of formulas exact up to degree D for integration with respect to the standard normal distribution. Such quadrature rules include the following:*

1. *Gauss–Hermite rule [17, Sec. 7.4]. The nodes of the Gauss–Hermite quadrature formula of order ν $(\nu = 1, 2, \dots)$ are the roots of the* Hermite polynomial *of degree ν, defined by $H_\nu(x) = (-1)^\nu e^{x^2/2} \frac{d^\nu}{dx^\nu} e^{-x^2/2}$. With appropriately chosen weights this formula is exact for all polynomials up to degree $2\nu - 1$, which is the highest possible degree for formulas with ν nodes. The Gauss–Hermite rule is not nested.*

2. *Genz–Keister rule [12]. The Genz–Keister rule is obtained by a straightforward adaptation of Patterson's algorithm [24] that yields the GKP rule for the uniform distribution. This rule defines a sequence of nested quadrature formulas for the standard normal distribution. Similarly to the GKP sequence, it is finite: the number of nodes of its first five formulas are $1, 3, 9, 19$, and 35, the corresponding degrees of exactness are $1, 5, 15, 29$, and 51.*

The transformation of formulas via the diffeomorphism g is also the way to apply QMC sampling for general distributions: in Step 1 of the above algorithm the scenarios $\mathcal{H}(q, n)$ can be replaced by the points of a low-discrepancy sequence to obtain the QMC formulas.

3.2 Sparse grid scenarios using Patterson-type quadrature rules

The discussion in Example 1 can be applied to distributions other than the uniform and the normal distributions. *Gaussian quadrature formulas*, that is, formulas with degree $2\nu - 1$ of polynomial exactness with $L(\nu) = \nu$ nodes can be obtained for every continuous distribution [17, Chap. 7]. These formulas are unique (for given ν and distribution), and they have the highest possible degree of polynomial exactness, but they are not nested.

It is also possible to generalize Patterson's method from [24] that yields the GKP rule for the uniform distribution to obtain nested sequences of quadrature formulas for other continuous distributions with finite moments. One such extension is given in [25]; a streamlined version, which computes nested sequences of quadrature formulas for general continuous distributions with finite moments directly from the moments of the distribution, can be found in [18].

4 Convergence of sparse grid scenario generation

We now give convergence results for the stochastic optimization problems (2) generated using the sparse grid scenarios. Theorem 3 states that for integrands that are polynomials after the diffeomorphism to product form, the approximation (2) using sparse grid scenario gives an exact representation of (1).

Theorem 4 presents a novel uniform convergence result for optimization using sparse grid approximations for functions with bounded weak derivatives. Note that since the sparse grid method may generate negative scenario weights, the previous convergence results by King and Wets [16], Donohue [8], and Pennanen [26] do not apply, and also that convergence result in Theorem 4 is slightly stronger than the epi-convergence results for the QMC methods in [26]. As the proof relies on Theorem 2, the result requires that the function $f(g(\cdot))$ has bounded weak derivatives. Since g is differentiable, this holds for smooth functions f, except for some cases when the domain Ξ is unbounded. In such cases suitable truncation of the underlying distribution (and Ξ) may be used. If the distribution is a linear transform of a product of univariate distributions (with bounded or unbounded support), then the method of Section 3.2 is applicable.

Finally, a rate of convergence result for the sparse grid approximation is given in Theorem 5.

Theorem 3. *Consider the optimization problem* (1) *and its approximation* (2)*, where the weights and scenarios are generated using the sparse grid method of Section 3.1, with some control parameter $q \geq 1$. If the underlying univariate quadrature rule used in the sparse grid construction has degree D_{ν_i} of polynomial exactness at resolution ν_i, and $\forall x \in \mathcal{C}$ the function $f(g(\cdot), x)$ is a member of the space of polynomials $\mathbb{P}_q^n$ defined in Theorem 1, then x^* is an optimal solution of (1) if and only if x^* is an optimum solution of (2).*

Proof. Follows immediately from Theorem 1: under the assumptions, the approximation (2) is exact, the two problems are identical. □

Theorem 4 (Convergence of the sparse grid method)**.** *Consider (1) and assume that $\mathcal{C}$ is closed and bounded; $f(\xi, x)\rho(\omega) \leq M < \infty$ for all $x \in \mathcal{C}$ and $\xi \in \Xi$; and that $\forall x \in \mathcal{C}$, $f(g(\cdot), x)$ is in the space $\mathcal{W}_n^r$, $1 \leq r < \infty$. Consider (2) where the scenarios are generated using the sparse grid method of Section 3. Let x^K be a solution of (2), $K = 1, \ldots, \infty$, and z_K^* be the corresponding objective value. Then,*

(i) $z^* \geq \overline{\lim}_K z_K^*$.

(ii) If $\{x^K\}_{K=1}^{\infty}$ has a cluster point $\hat{x}$, then $\hat{x}$ is an optimal solution of (1). Furthermore, for a subsequence $\{x^{K_t}\}_{t=1}^{\infty}$ converging to $\hat{x}$, $\lim_t z_{K_t}^* \to z^*$.

Proof. Let $F(x) = \int_\Xi f(\xi, x)P(d\xi)$ and $Q_K(x) = \sum_{k=1}^K w^k f(\xi^k, x)$. From our earlier discussion $F(x) = \int_{\Omega^n} f(g(\omega), x)\mu(d\omega)$. Since $f(g(\cdot), x) \in \mathcal{W}_n^r, \forall x \in \mathcal{C}$, by Theorem 2 we have that for every $x \in \mathcal{C}$,

$$\begin{aligned} |F(x) - Q_K(x)| &\leq c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} \|f(\cdot, x)\|_\infty \\ &\leq c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} M. \end{aligned}$$

Hence, as $K \to \infty$, $Q_K(x) \to F(x)$ uniformly on $\mathcal{C}$, i.e., for every convergent sequence $\{x^K\}_{k=1}^{\infty} \to \bar{x} \in \mathcal{C}$, we have

$$\forall \epsilon > 0, \exists N_1(\epsilon) : |Q_n(x^K) - F(x^K)| < \epsilon, \forall n > N_1(\epsilon). \tag{10}$$

Since F is continuous on the closed and bounded set $\mathcal{C}$, it is uniformly continuous there, and we have

$$\exists \beta(\epsilon) > 0 : |F(x^K) - F(\bar{x})| < \epsilon \text{ if } \|x^K - \bar{x}\| < \beta(\epsilon). \tag{11}$$

If $x^K \to \bar{x}$, the above inequality holds for all K large enough, say $K \geq N_2(\epsilon)$. Hence, we have

$$|F(\bar{x}) - Q_K(x^K)| \leq |F(\bar{x}) - F(x^K)| + |F(x^K) - Q_n(x^K)| < 2\epsilon, \forall K > \max(N_1(\epsilon), N_2(\epsilon)). \tag{12}$$

Therefore, $Q_K(x^K) \to F(\bar{x})$. Clearly,

$$Q_K(x^K) \geq \inf_{x \in \mathcal{C}} Q_K(x), \tag{13}$$

hence,

$$F(\bar{x}) = \lim_K Q_K(x^K) = \overline{\lim_K} Q_K(x^K) \geq \overline{\lim_K} \left(\inf_{x \in \mathcal{C}} Q_K(x) \right). \tag{14}$$

By taking infimum of the above inequality, we have

$$z^* = \inf_{x \in \mathcal{C}} F(\cdot) \geq \overline{\lim_K} \left(\inf_{x \in \mathcal{C}} Q_K(x) \right) = \overline{\lim_K} z_K^*. \tag{15}$$

We now prove the second result. The solution sequence $\{x^K\}_{K=1}^{\infty}$ of the problem sequence $(2)_{K=1}^{\infty}$ might not converge. However if it has a cluster point $\hat{x}$, we consider a subsequence $\{x^{K_t}\}_{t=1}^{\infty} \to \hat{x}$. By applying (14) to this subsequence, we have

$$F(\hat{x}) = \lim_t Q_{K_t}(x^{K_t}) = \lim_t \left(\inf_{x \in \mathcal{C}} Q_{K_t}(x) \right), \tag{16}$$

and since

$$F(\hat{x}) \geq \inf_{x \in \mathcal{C}} F(x), \tag{17}$$

we have

$$\overline{\lim_t} \left(\inf_{x \in \mathcal{C}} Q_{K_t}(x) \right) \leq \inf_{x \in \mathcal{C}} F(x) \leq F(\hat{x}) = \underline{\lim_t} \left(\inf_{x \in \mathcal{C}} Q_{K_t}(x) \right). \tag{18}$$

Hence, $\hat{x}$ is an optimal solution of (1), and $\lim_t z_{K_t}^* = z^*$. □

Theorem 5 (Rate of convergence)**.** *Consider (1) and its sparse grid approximation (2). Assume that $\forall x \in \mathcal{C}$ the function $f(g(\cdot), x) \in \mathcal{W}_n^r$, and $f(\xi, x)$ is bounded for all $x \in \mathcal{C}$. Let x^* be an optimal solution of (1), and x^K be an optimal solution of (2), then*

$$\left| \sum_{k=1}^{K} w^k f(\xi^k, x^K) - \int_\Xi f(\xi, x^*) P(d\xi) \right| \leq \epsilon, \text{ and} \tag{19}$$

$$\left| \int_\Xi f(\xi, x^K) P(d\xi) - \int_\Xi f(\xi, x^*) P(d\xi) \right| \leq 2\epsilon, \tag{20}$$

where

$$\epsilon = c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} \|f\|_\infty, \tag{21}$$

and K is defined as in Theorem 2.

Proof. Let $F(x) = \int_\Xi f(\xi, x) P(d\xi)$ and $Q_K(x) = \sum_{k=1}^{K} w^k f(\xi^k, x)$. Then from Theorem 2, and using the optimality of x^K and x^*, we have

$$F(x^*) - Q_K(x^K) = F(x^*) - F(x^K) + F(x^K) - Q_K(x^K) \leq F(x^K) - Q_K(x^K) \leq \epsilon, \text{ and}$$
$$F(x^*) - Q_K(x^K) = F(x^*) - Q_K(x^*) + Q_K(x^*) - Q_K(x^K) \geq F(x^*) - Q_K(x^*) \geq -\epsilon,$$

for K sufficiently large. This proves (19). Now from (19) and Theorem 2, respectively, we have

$$-\epsilon \leq Q_K(x^K) - F(x^*) \leq \epsilon$$
$$-\epsilon \leq F(x^K) - Q_K(x^K) \leq \epsilon.$$

Hence, $-2\epsilon \leq F(x^K) - F(x^*) \leq 2\epsilon$. □

Theorem 5 suggests that using the sparse grid approximation the optimal objective value of (2) converges to the optimal objective value of (1) with the same rate as that for the integration problem. Hence, it continues to combat the curse of dimensionality for stochastic optimization problems involving differentiable functions.

5 Numerical Examples

Our first example (Section 5.1) demonstrates the finite convergence of the sparse grid scenario generation for polynomial models. We consider a simple example involving the Markowitz model, taken from Rockafellar and Uryasev [30]. There are three instruments: S&P 500, a portfolio of long-term U.S. government bonds, and a portfolio of small-cap stocks, the returns are modeled by a joint normal distribution. The objective, the variance of the return of the portfolio, can be expressed as the integral of a quadratic polynomial.

In Section 5.2 we consider a family of utility maximization problems. We consider three different utility functions and three different distributions: normal, log-normal, and one involving Beta distributions of various shapes. With these examples we examine the hypothesis that with a sufficiently smooth integrand sparse grid formulas with high degree of exactness provide a good approximation of the optimal solutions to stochastic programs even for high-dimensional problems, and also regardless of the shape of the distribution with respect to which is integrated.

In the examples we compare variants of the sparse grid method to Monte Carlo (MC) and quasi-Monte Carlo sampling. Our preliminary experiments with various quasi-random (or low-discrepancy) sequences in the QMC method showed relatively little difference between different quasi-random sequences. We show this similarity in the first example, but for simplicity, in the utility maximization problems only the results with the Sobol sequence are shown. (For more examples see the earlier technical report version of the paper.)

The results are summarized in Section 5.3.

In our experiments we used the GKP univariate quadrature rules to build sparse grid formulas for the uniform distribution, and analogous nested quadrature rules to be used with non-uniform distributions. The code to generate multivariate scenarios from Smolyak sparse grid points was written in Matlab; the approximated problems were solved with Matlab's `fmincon` solver.

5.1 Markowitz model

This instance of the classic Markowitz model was used in [30] and [28] for comparing QMC scenario generation methods with the Monte Carlo method; the problem data is reconstructed in the Appendix.

Example 2. *(Markowitz model) Let $x = [x_1, \dots, x_n]$ be the amount invested in d financial instruments, $x_i \geq 0$ and $\sum_{i=1}^{n} x_i = 1$. Let $\xi = [\xi_1, \dots, \xi_n]$ be the random returns of these instruments, $p(\xi)$ be the density of the joint distribution of the rates of return, which is a multinormal distribution with mean vector m and covariance matrix $V \in \mathbb{R}^{n \times n}$. We require that the mean return of the portfolio x be at least R, and we wish to minimize the variance of the portfolio. The problem can be formulated as follows:*

$$\begin{aligned} &\min_x \int_{\mathbb{R}^n} (\xi^T x - m^T x)^2 p(\xi) d\xi \\ &\text{s.t. } \|x\|_1 \leq 1, m^T x \geq R, x \geq 0. \end{aligned}$$

We can approximate the above problem by a formulation with finitely many scenarios as

$$\begin{aligned} &\min_x \sum_{k=1}^{K} w_k (\xi_k^T x - m^T x)^2 \\ &\text{s.t. } \|x\|_1 \leq 1, m^T x \geq R, x \geq 0, \end{aligned}$$

using scenarios x_k and weights w_k.

The optimal objective function value can also be computed by solving an equivalent (deterministic) quadratic programming problem; the optimal value is approximately 0.003785.

The objective values of the approximated models are shown in Table 2 and plotted in Figure 2. Since the integrand is a quadratic polynomial, approximation using a sparse grid formula with degree of polynomial exactness greater than one (for integration with respect to the normal distribution) is guaranteed to give the exact optimum. For example, the convergence of the sparse grid method using the Genz–Keister quadrature rule is *finite*: we obtain the exact objective function value with 7 scenarios, corresponding to the control parameter value $q = 2$ in the sparse grid construction (6), which yields an exact formula for polynomial integrands of degree up to 3. (The formula corresponding to $q = 1$ is exact only for polynomials of degree one.)

We can also use the sparse grid method with GKP nodes transformed using the diffeomorphism mapping uniformly distributed random variables to normally distributed ones. Of course, the convergence for such formulas is not finite. We obtain 4 correct significant digits for sample sizes exceeding 1023, which corresponds to $q = 6$ in (6). As noted before, using the GKP formulas requires care: the diffeomorphism required to transform the uniform distribution to normal does not have bounded derivatives, thus the convergence results do not apply. One possibility to resolve this problem is to replace the normal distribution by a sufficiently truncated one.

We also generated QMC approximations with the same number of scenarios for this problem, these results are also reported in Table 2. The performance of the Sobol, Halton, and reverse Halton sequences are comparable to each other. Their performance is better than the Niederreiter-2 sequence, and significantly better than the Monte Carlo based approximations. The results show that the sparse grid method reached the first, second, third, and fourth significant digit at 7, 111, 351, and 1023 number of scenarios. In contrast, the best performing QMC sequences needed 111 and 1023 number of scenarios to reach the one and two significant digits of accuracy. Note also that with 2815 scenarios the standard deviation computed using MC scheme only provides confidence for the first significant digit of the objective value for this problem. In order to study the convergence of QMC for this problem, we generated approximations with up to $500,000$ nodes using the Sobol sequence. We observed steady but slow convergence. The third correct significant digit was obtained with $2,300$ scenarios, and the fourth correct significant digit was achieved with $110,000$ scenarios.

It is reasonable to conclude that by exploiting the smoothness of the integrand function the sparse grid method achieved faster convergence than the scenarios generated using popular QMC sequences even using the transformed GKP formula, which does not give exact result for a polynomial integrand.

5.2 Utility maximization models

In this section we examine the hypothesis that with a sufficiently smooth integrand sparse grid formulas with high degree of exactness provide a good approximation of the optimal solutions to stochastic programs even for high-dimensional problems, and also regardless

	sparse grid		QMC				MC	
# nodes	GKP	Genz–Keister	Sobol	Nieder.	Halton	RevHalton	mean	std
1	0.000000	0.000000	0.000000	0.115447	0.000085	0.000085	0.0025546	0.0038893
7	0.003091	0.003785	0.001099	0.017470	0.001695	0.002253	0.0038511	0.0022252
31	0.003674		0.002990	0.006398	0.002875	0.003346	0.0033804	0.0008580
111	0.003769		0.003398	0.004499	0.003408	0.003543	0.0038163	0.0006048
351	0.003783		0.003641	0.004023	0.003640	0.003683	0.0037725	0.0002422
1023	0.003785		0.003726	0.003840	0.003737	0.003737	0.0038138	0.0002272
2815	0.003785		0.003760	0.003802	0.003759	0.003767	0.0037926	0.0001009

Table 2: Approximated Objective Value of the Markowitz Model. The true optimal objective value is ≈ 0.003785.

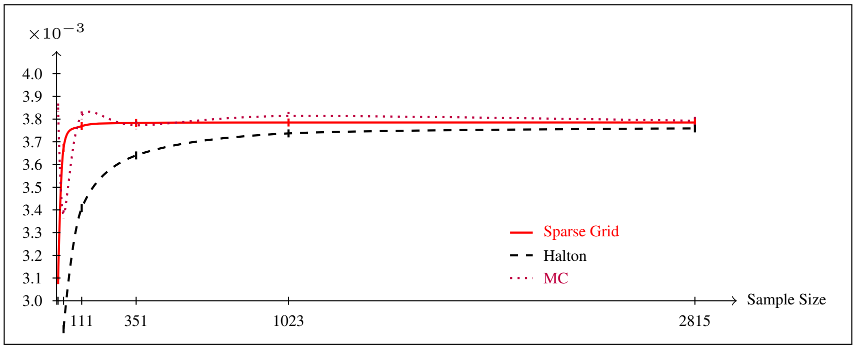


Figure 2: Approximated objective value of the Markowitz Model. The y-axis shows the optimal values from the approximated model for different scenario generation methods for sample size (x-axis) from 1 to 2815. The true optimal value is ≈ 0.003785.

of the shape of the distribution with respect to which it is integrated. For this purpose, we considered utility maximization examples of the form

$$\max_x \int_\Xi u(x^\mathrm{T}\xi)p(\xi)d\xi \quad \text{s.t.} \quad \|x\|_1 \leq 1, x \geq 0, \tag{22}$$

for different utility functions u and density functions p.

The three utility functions considered were:

$$u_1(t) = -\exp(t) \quad \text{(exponential utility)}, \tag{23a}$$

$$u_2(t) = \log(1+t) \quad \text{(logarithmic utility), and} \tag{23b}$$

$$u_3(t) = (1+u)^{1/2} \quad \text{(power utility)}. \tag{23c}$$

The probability densities considered were products Beta distributions, obtained by taking the product of univariate Beta(α,β) distributions with $\alpha, \beta \in \{1/2, 1, 3/2, 5\}$ (see Figure 3). The motivation behind this choice that it allows us to experiment with distributions of various shapes, and also to transform the problem into product form, for which

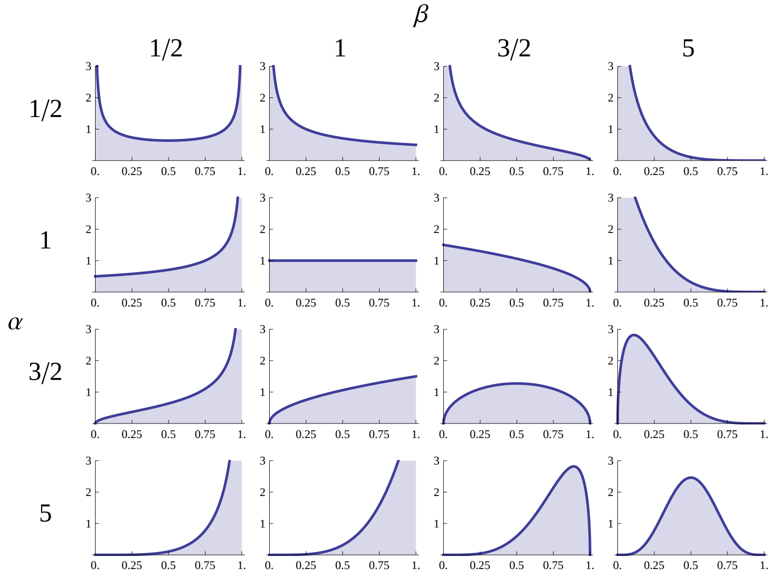


Figure 3: Shapes of Beta(α,β) distributions for $(\alpha, \beta) \in \{1/2, 1, 3/2, 5\}^2$.

formulas with different degrees of polynomial exactness can be created and compared. We compared both variants of the sparse grid method proposed in Section 3: we used GKP formulas transformed with the appropriate diffeomorphism to scenarios for integration with respect to the product Beta distribution (or *transformed GKP* formulas for short) and sparse grid formulas using the Patterson-type quadrature rules of Section 3.2 derived for the Beta distribution (or *Patterson-type sparse grid* for short).

5.2.1 Exponential utility

Table 3 shows the (estimated) optimal objective function values computed with different scenario generation techniques for the 100-dimensional exponential utility maximization (using u_1 from (23) in (22)), where p is the probability distribution function of the 100-fold product of the Beta(1/2,1/2) distribution (shown in the upper left corner of Figure 3). The optimal objective function value cannot be computed in a closed form, but by the agreement of the numerical results obtained with the different methods the correct value (up to six significant digits) appears to be 0.606909. The table shows great difference in the rate of convergence of the different scenario generation methods. Monte Carlo (MC) integration achieves only 4 correct significant digits using 10^6 scenarios, quasi-Monte Carlo (QMC) integration with the Sobol sequence gets 6 digits with about $2 \cdot 10^5$ scenarios, but only 3 digits with $2 \cdot 10^4$ scenarios. The sparse grid formula obtained by transforming the GKP rule gets 6 digits with $2 \cdot 10^4$ scenarios. (See column 4 in Table 3.) Finally, the Patterson-type sparse grid achieves 6 correct digits already with 201 scenarios. (Column 5.) The latter formula was created using a nested Patterson-type rule for the Beta($1/2$,$1/2$) distribution

# nodes	MC	QMC (Sobol)	SG w/ transformed GKP	Patterson-type SG
201	0.5861458741	0.6057992465	0.6069012301	0.6069097420
20401	0.6059951776	0.605986365	0.6069098645	0.6069098767
200000	0.606784018	0.6069097597		
1000000	0.6069217022	0.6069097569		

Table 3: Results from a 100-dimensional exponential utility maximization example using Beta distributions.

(see Section 3.2).

In summary, to achieve the same accuracy in this example, the sparse grid using the Patterson-type rule (and which is exact for polynomials) requires an order of magnitude fewer scenarios than the sparse grid using transformed GKP nodes, which in turn requires an order of magnitude fewer scenarios than QMC. Monte Carlo is not competitive with the other three methods.

We repeated the same experiment with all of the 16 distributions shown on Figure 3, with the same qualitative results, with the exception of the distribution Beta(1,1), which is the uniform distribution, hence the two sparse grid formulations are equivalent (and still outperform MC and QMC); the details are omitted for brevity.

We also considered examples with less regular distributions, using the same distributions from Figure 3 as components. The underlying 160-dimensional product distribution has ten components distributed as each of the distributions shown on Figure 3. The optimal objective function value appears to be approximately 0.40315 by the agreement of three out of four methods; Table 4 shows the approximate objective function values computed with different techniques. The sparse grid formula using the Patterson-type rule achieves the same precision as QMC with an order of magnitude fewer points, reaching five correct digits with only 341 nodes. MC performs considerably worse than both of them, it needs about 1 million points to get the fourth significant digit correctly. Memory constraints prevented the generation of higher-order sparse grids formulas, the transformed GKP formula did not outperform MC in this example.

5.2.2 Logarithmic utility

We repeated the above experiments for the logarithmic utility maximization problem, that is, plugging u_2 from (23) into (22), with the same experimental setup. We only give here the detailed results of the 160-dimensional experiment involving the product Beta distribution with various parameters. The optimal objective function value appears to be approximately -0.646451. The results obtained with different scenario generation methods are shown in Table 5.

The results are essentially the same as in the exponential utility maximization problem. Sparse grid with the Patterson-type quadrature rule gets 6 correct significant digits with 341 nodes, whereas QMC requires about 10^6 nodes for the same accuracy.

# nodes	MC	QMC (Sobol)	SG w/ transformed GKP	Patterson-type SG
321	0.4000670523	0.4026883669	0.4007432411	
341	0.4020710589	0.4027479863		0.4031483327
5000	0.4031330914	0.4031128470		
51841	0.4025879651	0.4031793283	0.4028398967	
58331	0.4031793283	0.4031448802		0.4031484071
250000	0.4031696956	0.4031478092		
1000000	0.4031451156	0.4031482230		

Table 4: Results from the 160-dimensional exponential utility maximization example using Beta distributions.

# nodes	MC	QMC (Sobol)	SG w/ transformed GKP	Patterson-type SG
321	-0.6507230838	-0.6471053734	-0.6493579499	
341	-0.6497278246	-0.6470413764		-0.6464513847
5000	-0.6469930332	-0.6464964703		
51841	-0.6464701611	-0.6464576966	-0.6467992796	
58331	-0.6465609053	-0.6464554973		-0.6464512999
250000	-0.6464336612	-0.6464519084		
1000000	-0.6464269362	-0.6464514985		

Table 5: Results from the 160-dimensional logarithmic utility maximization example using Beta distributions.

# nodes	MC	QMC (Sobol)	SG w/ transformed GKP	Patterson-type SG
321	-1.384429683	-1.3821357312	-1.3835622039	
341	-1.3863232853	-1.3821077423		-1.3816379583
5000	-1.3822980733	-1.3816702317		
51841	-1.381717902	-1.3816421717	-1.3800861876	
58331	-1.381763212	-1.3816406975		-1.3816379399
250000	-1.3816977241	-1.3816382958		
1000000	-1.3816521898	-1.381638062		

Table 6: Results from the 160-dimensional power utility maximization example using Beta distributions.

5.2.3 Power utility

We repeated the above experiments for the power utility maximization problem, that is, plugging u_3 from (23) into (22), with the same experimental setup. The results obtained with different scenario generation methods are shown in Table 6; they are very similar to the results of the previous experiments. The optimal objective function value appears to be approximately -1.3816. MC needs over 50000 nodes to get to 4 digits of accuracy, the fifth digit is reached with 10^6 nodes. QMC reaches 5 correct significant digits with about 50000 nodes. Sparse grid with the Patterson-type quadrature rule requires only 341 nodes for the same accuracy.

5.3 Summary of numerical results

The numerical experiments provide a strong indication of both the strengths and limitations of sparse grid scenario generation in stochastic optimization. For problems where the underlying distribution can be transformed linearly to a product of univariate distributions, sparse grid scenarios generated using Patterson-type formulas are superior to standard MC and QMC scenario generation methods.

Sparse grid scenario generation also scales well: for most common distributions $2n+1$ scenarios are sufficient to achieve degree 3 of polynomial exactness. $O(n^2)$, respectively $O(n^3)$, scenarios provide degree 5, respectively degree 7, of polynomial exactness, which is sufficient for the good approximation of smooth functions, even in optimization problems with hundreds (and possibly thousands) of variables. When the integrand can be expressed as the product of a polynomial and a probability density function, approximation with sparse grid scenarios provides the exact optimum, which cannot be matched by other scenario generation methods. The generation of the sparse grid formulas (even those with millions of nodes in thousands of dimensions) is a minimal overhead compared to the optimization step. The generation of Patterson-type formulas with the method of [18] is also easy, and needs to be carried out only once for every univariate distribution used. The size of the problems concerned in this paper were only limited by two other factors: available memory for carrying out the optimization, and the fact that the true optimal objective value cannot be validated by alternative methods or statistical bounds for problems with thousands of random variables.

For problems where a general nonlinear diffeomorphism is needed to transform the problem to one involving a product distribution, the nonlinearity of the transformation eliminates the polynomial exactness of the formulas. Although the convergence of the method, as the number of scenarios tends to infinity, is proven in this case, too, the method did not outperform QMC sampling in our high-dimensional examples. Since the convergence of standard QMC methods is also independent of the smoothness of the integrand, it is recommended that QMC be used in place of sparse grid scenario generation for high-dimensional problems that cannot be written in product form.

6 Concluding Remarks

Sparse grid scenario generation appears to be a very promising alternative to quasi-Monte Carlo and Monte Carlo sampling methods for the solution of stochastic optimization problems, whenever the integrands in the problem formulation are sufficiently smooth. The theoretical results on its efficiency, which state that the rate of convergence of the optimal objective value is the same as the rate of convergence of sparse grid formulas for integration, is complemented by excellent practical performance on a variety of utility maximization problems, which feature the expected values of smooth concave utility functions as objectives.

It is well-known that sparse grid formulas using nested quadrature formulas achieve better approximation with approximately the same number of scenarios than those using non-nested, such as Gaussian, formulas. The numerical results also show the importance of using suitable univariate quadrature formulas, which allow the generation of scenarios that provide exact approximation for polynomial integrands up to some degree. Patterson-type quadrature formulas have both desirable properties. Patterson-type sparse grid scenarios provided consistently better approximations than those obtained through a non-linear (and non-polynomial) transformation of scenarios generated for a different (usually uniform) distribution. Scenarios with a given degree of polynomial exactness are easily generated for distributions that are linear transformations of a product of univariate distributions—this includes the multivariate normal distributions. We were able to generate univariate quadrature formulas of at least 5 different resolutions (with degrees of exactness exceeding 40) for a number of distributions. The limits of this approach is an open problem; for example it is not known if the sequence of GKP formulas is finite or not. Note that this theoretical gap is only relevant in the solution of low-dimensional problems. High-dimensional sparse grids of high resolution have prohibitively large number of nodes, furthermore, Gaussian formulas with arbitrarily high degree of polynomial exactness can be generated for every univariate distribution using established methods; these can also be used in the absence of Patterson-type formulas.

The sparse grid method also scales well. The scenarios can be generated quickly, and a small number of scenarios achieves a given low degree of polynomial exactness. The main limiting factor in the size of the problems considered in the numerical experiments section, while comparing sparse grid with the MC and QMC methods, was the size of the memory of the computer used for these experiments.

A couple of questions remain open, and shall be the subject of further study. The most important one concerns multi-stage problems. The first stage objective function of a multi-stage stochastic programming problem is typically the integral of a piecewise smooth, but not necessarily differentiable, convex function. The sparse grid approach is applicable in principle for many such problems (as long as the integrands involved have weak derivatives), but the practical rate of convergence may be slow, and the negative weights in the sparse grid formulas may make the approximation of the convex problems non-convex.

A Appendix: Problem data

Tables 7 and 8 are used in the Markowitz model; the minimum return R is 0.011. This problem is the same as the one used by Rockafellar and Uryasev in [30]. The high-dimensional Utility Maximization problems involved various Beta distributions, as explained in the main text.

Instrument	Mean Return
S & P	0.0101110
Gov Bond	0.0043532
Small Cap	0.0137058

Table 7: Portfolio Mean Return

	S & P	Gov Bond	Small Cap
S & P	0.00324625	0.00022983	0.00420395
Gov Bond	0.00022983	0.00049937	0.00019247
Small Cap	0.00420395	0.00019247	0.00764097

Table 8: Portfolio Covariance Matrix

References

[1] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[2] Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[3] M. Casey and S. Sen. The scenario generation algorithm for multistage stochastic linear programming. *Mathematics of Operations Research*, 30:615–631, 2005.

[4] G. Consigli, J. Dupačová, and S. Wallace. Generating scenarios for multistage stochastic programs. *Annal of Operations Research*, 100:25–53, 2000.

[5] Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, 1975.

[6] M.A.H. Dempster and R.T. Thompson. EVPI-based importance sampling solution procedure for multistage stochastic linear programming on parallel MIMD architectures. *Annal of Operations Research*, 90:161–184, 1999.

[7] C. Donohue. *Stochastic Network Programming And The Dynamic Vehicle Allocation Problem*. PhD thesis, The University of Michigan, Ann Arbor, 1996.

[8] Christopher J. Donohue and John R. Birge. The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. *Algorithmic Operations Research*, 1:20–30, 2006.

[9] J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming: An approach using probability metrics. *Mathematical Programming*, 95:493–511, 2003.

[10] Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.

[11] G. B. Folland. *Real Analysis: Modern Techniques And Their Applications*. Wiley, 2nd edition, 1999.

[12] Alan Genz and B.D. Keister. Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. *Journal of Computational and Applied Mathematics*, 71(2):299–309, 1996.

[13] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

[14] Holger Heitsch and Werner Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406, 2007.

[15] Michal Kaut and Stein W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Stochastic Programming E-Print Series*, 2003-14, 2003.

[16] A.J. King and R.J.-B Wets. Epi-convergency of convex stochastic programs. *Stochastic And Stochastic Reports*, 34:83–92, 1991.

[17] Vladimir Ivanovich Krylov. *Approximate Calculation of Integrals*. Dover, Mineola, NY, 2005.

[18] Sanjay Mehrotra and Dávid Papp. Generating nested quadrature formulas for general weight functions with known moments. Technical Report arXiv:1203.1554v1 [math.NA], Northwestern University, March 2012.

[19] Arnold Neumaier. *Introduction to Numerical Analysis*. Cambridge University Press, 2001.

[20] H. Niederreiter. *Random Number Generation And Quasi-Monte Carlo Methods*, volume 63. Society for Industrial and Applied Mathematics(SIAM), 1992.

[21] Harald Niederreiter and Denis Talay, editors. *Monte Carlo And Quasi-Monte Carlo Methods*. Springer-Verlag, 2006.

[22] Vladimir Norkin, Georg Pflug, and Andrzej Ruszczyński. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83:425–450, 1998. 10.1007/BF02680569.

[23] E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes,. *Numer. Math.*, 75:79–97, 1996.

[24] T. N. L. Patterson. The optimal addition of points to quadrature formulae. *Mathematics of Computation*, 22(104):847–856, 1968.

[25] T. N. L. Patterson. An algorithm for generating interpolatory quadrature rules of the highest degree of precision with preassigned nodes for general weight functions. *ACM Transactions on Mathematical Software*, 15:123–136, June 1989.

[26] T. Pennanen. Epi-convergent discretizations of multistage stochastic programs. *Mathematics of Operations Research*, 30:245–256, 2005.

[27] T. Pennanen and M. Koivu. Integration quadrature in discretization of stochastic programs. *Stochastic Programming E-Print Series*, 2002-11.

[28] Teemu Pennanen and Matti Koivu. Epi-convergent discretizations of stochastic programs via integration quadratures. *Numerische Mathematik*, 100:141–163, 2005. 10.1007/s00211-004-0571-4.

[29] G. Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89:251–271, 2001.

[30] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.

[31] Walter Rudin. *Real And Complex Analysis*. McGraw-Hill Science/Engineering/Math, 1986.

[32] S.A. Smolyak. Interpolation and quadrature formula for the class W_s^a and E_s^a. *Dokl. Akad. Nauk SSSR*, 131:1028–1031, 1960. (in Russian, English Translation: Soviet Math. Dokl. 4, 240-243 (1963)).

[33] Iiya M. Sobol. *A Primer for the Monte Carlo Method*. CRC Press, 1994.

[34] Stein W. Wallace and William T. Ziemba, editors. *Applications Of Stochastic Programming*. Society for Industrial and Applied Mathematics, 2005.

[35] Grzegorz W. Wasilkowski and Henryk Wozniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11(1):1–56, 1995.