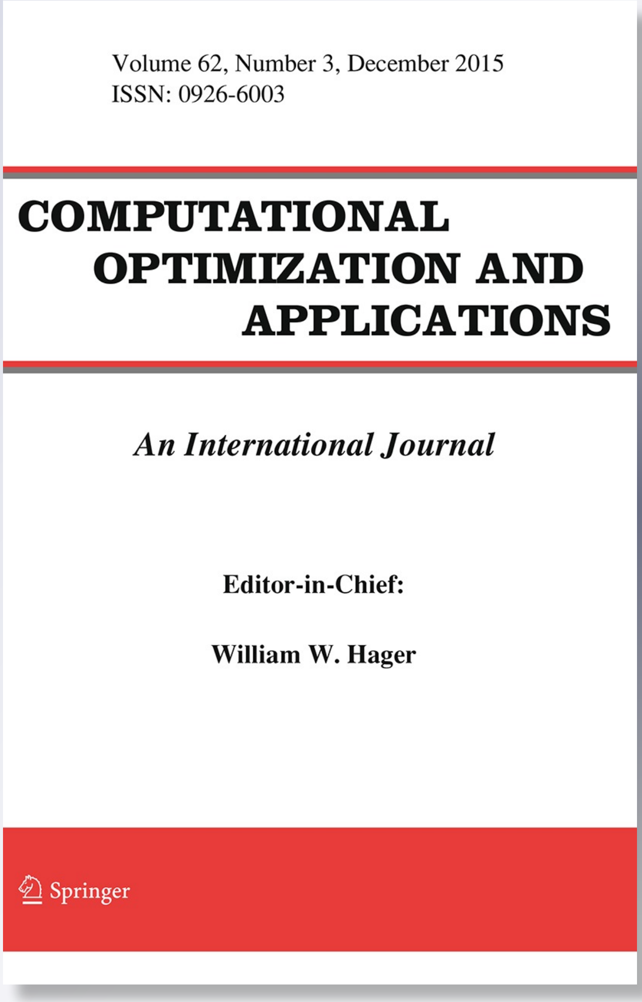
Scenario generation for stochastic optimization problems via the sparse grid method

Michael Chen, Sanjay Mehrotra & Dávid Papp

Computational Optimization and Applications
An International Journal

ISSN 0926-6003
Volume 62
Number 3

Comput Optim Appl (2015) 62:669-692
DOI 10.1007/s10589-015-9751-7



Volume 62, Number 3, December 2015
ISSN: 0926-6003

COMPUTATIONAL OPTIMIZATION AND APPLICATIONS

An International Journal

Editor-in-Chief:

William W. Hager

Springer

Comput Optim Appl (2015) 62:669–692
DOI 10.1007/s10589-015-9751-7

Scenario generation for stochastic optimization problems via the sparse grid method

Michael Chen[1] · Sanjay Mehrotra[2] · Dávid Papp[3]

Received: 22 October 2012 / Published online: 19 April 2015

Abstract We study the use of sparse grids in 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 involved, the sequence of optimal objective function values of the sparse grid approximations converges to the true optimal objective function values as the number of scenarios increases. The rate of convergence is also established. We treat separately the special case when the underlying distribution is an affine transform of a product of univariate distributions, and show how the sparse grid method can be adapted to the distribution by the use of quadrature formulas tailored to the distribution. We numerically compare the performance of the sparse grid method using different quadrature rules with classic quasi-Monte Carlo (QMC) methods, optimal rank-one lattice rules, 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, especially 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. It is indicated that the method scales well with the dimension of the distribution—especially when the underlying distribution is an affine transform of a product of univariate distributions, in which case the method appears scalable to thousands of random variables.

✉ Dávid Papp
dpapp@ncsu.edu

[1] Department of Mathematics and Statistics, York University, Toronto, Canada

[2] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, USA

[3] Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

Keywords Scenario generation · Stochastic optimization · Discretization · Sparse grid

1 Introduction

Stochastic optimization is a fundamental tool in decision making, with a wide variety of applications (see, e.g., Wallace and Ziemba [43]). 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, 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 probability measure P is often given or approximated by a discrete one, with finite number of scenarios $\xi^k (k = 1, \ldots, K)$ with positive probabilities w^k, which amounts to the approximation of the integral in (1) by a finite sum:

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

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 [36] use *quasi-Monte Carlo (QMC) methods*, that is, classic low-discrepancy sequences, the Faure sequence, the Sobol sequence, or the Niederreiter sequence, as scenarios along with uniform weights. The QMC literature has undergone a tremendous improvement in the last decade or so, with the development of more advanced number-theoretic rules, digital nets, and lattice rules. Summarizing all the developments here is impossible; the interested reader is encouraged to consult one of the recent comprehensive works on QMC methods, such as [7].

Scenario reduction methods, e.g [4,16] 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 [10,38] choose the scenarios by minimizing a probability metric; the same approach is also applicable to scenario reduction. Casey and Sen [2] apply linear programming sensitivity analysis to guide scenario generation. Finally, the authors of this paper have previously approached scenario generation in data-driven settings by *moment matching*, that is, matching or approximating the sample moments of observed scenarios [25].

King and Wets [19] and Donohue [8] studied epi-convergence of the MC method. Epi-convergence of the QMC methods is established in Pennanen and Koivu [37], and Pennanen [35]. Since establishing the rate of convergence is difficult, different scenario generation methods are usually compared to each other numerically. Pennanen and Koivu [37] 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 MC sampling (see, e.g, [30]) may be used to compare scenario generation methods. Kaut and Wallace [18] give further guidelines for evaluating scenario generation methods.

The main contributions of this paper are twofold. Two variants of a sparse grid scenario generation method are proposed for the solution of stochastic optimization problems. Neither method requires knowledge of the exact distribution; it is sufficient that the moments up to a high enough order are known. However, to simplify the presentation we shall always assume that we have a known distribution. After establishing their convergence and rate of convergence (Theorems 4, 5), we numerically compare the two sparse grid methods 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 MC and QMC methods for smooth integrands, especially when the uncertainty is expressible using distributions that are affine transformations of a product of mutually independent 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 Sect. 2 we review Smoljak's sparse grid approximation method and its application to scenario generation. We present two variants of the method. A general method is presented in Sect. 2.3, whereas the variant presented Sect. 2.4 is designed for problems that can be transformed affinely to a problem whose underlying probability distribution is a product of univariate distributions.

In Sect. 2.5 we show that both variants of the sparse grid method give an exact representation of (1) if the integrand function belongs to a certain polynomial space. In Sect. 3 we also show that the first variant of the method is uniformly convergent under appropriate assumptions, and that its rate of convergence is the same as the rate of convergence of the underlying sparse grid numerical integration methods. Numerical results follow in Sect. 4. We give results both on stochastic optimization problems which have been used to compare scenario generation methods 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 MC and QMC methods for smooth integrands, and scales very well to problems with a large number of random variables when the underlying distribution is an affine transformation of a product of multivariate distributions. The results are summarized in Sect. 5.

2 Sparse grid methods in numerical integration

2.1 Univariate quadrature rules

The basic ingredient of sparse grid scenario generation methods is a univariate *quadrature rule*. A quadrature rule for a weight function ρ gives, for every $\nu \in \mathbb{N}$, a set of *points* (or *nodes*) $\{\omega_{\nu,1}, \dots, \omega_{\nu,L(\nu)}\} \subseteq \mathbb{R}^{L(\nu)}$ and corresponding *weights* $w_{\nu,1}, \dots, w_{\nu,L(\nu)} \in \mathbb{R}$, used to approximate a one-dimensional integral:

$$\int_{\Omega} g(\omega)\rho(\omega)d\omega \approx I_\nu[g] \overset{\text{def}}{=} \sum_{k=1}^{L(\nu)} w_{\nu,k} g(\omega_{\nu,k}). \tag{3}$$

Here 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*. Equivalently, we can identify a quadrature rule with the sequence of functionals $I_1[\cdot], I_2[\cdot], \dots$. Naturally, both the nodes and the weights depend on the given weight function ρ and the domain Ω, although for the sake of simplicity we shall not emphasize this in our notation; both of them will always be clear from the context. For the stochastic optimization problems of our concern it is sufficient to consider probability density functions ρ 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 points is specified by the function $L(\nu) : \mathbb{N} \to \mathbb{N}$, where ν is called the *resolution* of the formula. Different univariate quadrature rules differ in $L(\cdot)$, and in the nodes and weights they prescribe 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 [27] 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 [20].

We mention two particularly important families of quadrature rules that are also used in this paper. *Gaussian quadrature* rules [20, Chap. 7] are the generalization of Gauss–Legendre rules, and can be defined for every domain and weight function; they satisfy $L(\nu) = \nu$. We introduce the second family, *Patterson-type rules* in Sect. 2.4; the number of points in these rules increase exponentially with the resolution.

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 every density function ρ there exists a unique quadrature rule (known as Gaussian rule) satisfying $L(\nu) = \nu$ and $D_\nu = 2\nu - 1$ for every $\nu \geq 1$; see, for example, [20]. As another example, the GKP rule (for the uniform distribution) satisfies $D_1 = 1$ and $D_\nu = 3 \cdot 2^{\nu-1} - 1$, for $\nu \geq 2$ [14]. 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 the error in the approximation (3) can be estimated as:

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

where $K = L(\nu)$ is the number of nodes. (See also Theorem 2 below).

If the nodes Ω_ν of a quadrature rule satisfy $\Omega_\nu \subset \Omega_{\nu+1}$ for every ν (that is, if every node at a given resolution is also among the nodes of all higher resolution node sets), then the univariate quadrature rule is said to be *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 GKP formulas [14], as well as the Patterson-type rules of Sect. 2.4. In general, nested quadrature rules are preferred in the sparse grid scenario generation, as sparse grids built using nested rules have fewer scenarios than those built using non-nested rules. (This follows immediately from the construction presented in the next section.) Nested quadrature rules for non-uniform distributions can be generated, for example, using the algorithm described in [24].

It follows from the definition that a univariate quadrature formula with polynomial exactness of degree d (with respect to the distribution P) has the same polynomial exactness for every distribution with the same first d moments as P. In other words, the construction of univariate quadrature formulas do not need an exact specification of P, it is sufficient to know enough moments of P. The same holds for multivariate distributions, and specifically, the sparse grid formulas.

2.2 Product grids, and Smoljak's sparse grid construction for $[0, 1]^n$

If $\Xi = \Omega^n = [0, 1]^n$ and $\rho(\omega_1, \ldots, \omega_n) = \rho_1(\omega_1) \cdots \rho_n(\omega_n)$ is a separable function, simple formulas for approximating the integral $\int_\Xi f(\xi)\rho(\xi)d\xi$ can be obtained by considering direct products of univariate quadrature formulas, as follows.

For each $i = 1, \ldots, n$, fix a quadrature rule I^i; this gives a formula I^i_ν at resolution ν, with nodes $\omega^i_{\nu,1}, \ldots, \omega^i_{\nu,L_i(\nu)}$ and corresponding weights $w^i_{\nu,1}, \ldots, w^i_{\nu,L_i(\nu)}$. In other words, let the univariate quadrature rule for the weight function ρ_i be given as the sequence of functionals

$$I^i_\nu[g] = \sum_{k=1}^{L_i(\nu)} w^i_{\nu,k} g\left(\omega^i_{\nu,k}\right); \qquad \nu = 1, 2, \ldots.$$

Then define, for every multi-index $\nu = (\nu_1, \ldots, \nu_n) \in \mathbb{N}^n$, the *product* of the univariate formulas at *resolution* ν as

$$\left(I^1_{\nu_1} \otimes \cdots \otimes I^n_{\nu_n}\right)[f] \overset{\text{def}}{=} \sum_{k_1=1}^{L_1(\nu_1)} \cdots \sum_{k_n=1}^{L_n(\nu_n)} w^1_{\nu_1,k_1} \cdots w^n_{\nu_n,k_n} f\left(\omega^1_{\nu_1,k_1}, \ldots, \omega^n_{\nu_n,k_n}\right).$$

We call the resulting multivariate quadrature rule the *product* of the underlying univariate quadrature rules, and the use of this formula for multivariate numerical integration is sometimes referred to as the “full grid” or “dense grid” approach, as opposed to the “sparse grid” approach discussed below.

The degree of exactness of the above product formula is the same as the minimum of the exactness of the underlying univariate formulas. It is clear that this approach does not scale well with the dimension; Table 1 shows the number of nodes of the products of univariate Gaussian quadrature rules for different dimensions and degrees of exactness.

Sparse grid formulas are also based on univariate formulas and their products, but they avoid the sharp increase in the number of nodes as the dimension increases by using a combination of lower resolution formulas. The classic sparse grid approximation framework was established by Smoljak in [41]; to keep the paper self-contained, we now briefly describe this construction applied to multivariate integration, and summarize the main results on this topic. See also [1] for a more comprehensive review on sparse grid methods (with a focus on differential equations), and [15] for a short and transparent introduction to its application to numerical integration.

As above, let $\rho(\omega_1,\dots,\omega_n)=\rho_1(\omega_1)\cdots\rho_n(\omega_n)$ be a given separable n-variate weight function; we shall construct a multivariate quadrature formula for every given resolution $\nu=(\nu_1,\cdots,\nu_n)$. Let $\Omega^i_{\nu_i}=\{\omega^i_{\nu_i,1},\dots,\omega^i_{\nu_i,L_i(\nu_i)}\}$ be the nodes of the univariate quadrature rule for the weight function ρ_i with resolution ν_i, and let $w^i_{\nu_i,1},\dots,w^i_{\nu,L_i(\nu_i)}$ be the corresponding weights.

Now, for every given positive integer q, Smoljak's sparse grid formula uses the points in

$$\mathcal{G}(q,n)\overset{\text{def}}{=}\bigcup_{q\le\|\nu\|_1\le q+n-1}\left(\Omega^1_{\nu_1}\times\cdots\times\Omega^n_{\nu_n}\right), \tag{4}$$

and approximates the multivariate integral

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

by a linear combination of smaller product formulas, given by

$$SG_q[f]\overset{\text{def}}{=}\sum_{q\le\|\nu\|_1\le q+n-1}(-1)^{q+n-\|\nu\|_1-1}\begin{pmatrix} n-1\\ \|\nu\|_1-q\end{pmatrix}\left(I^1_{\nu_1}\otimes\cdots\otimes I^n_{\nu_n}\right). \tag{6}$$

We now present results on the error estimates for the sparse grid method in approximating a multidimensional integral. Theorem 1 shows that Smoljak'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 n-variate polynomials of degree 3. A general result on polynomial exactness is restated in the following theorem.

Theorem 1 (Gerstner and Griebel [14], Novak and Ritter [31,32]) *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 $k_i \le D_{\nu_i}$ and $k_j \le D_{\nu_j}$. Then the value of (6) is equal to the integral (5) for every polynomial $f \in \mathbb{P}_q^n := \sum_{\|\nu\|_1 \le q+n-1} \left\{ \mathcal{P}_{D_{\nu_1}} \otimes \cdots \otimes \mathcal{P}_{D_{\nu_n}} \right\}$.*

The same holds if we replace the domain of the univariate quadrature rules $\Omega = [0, 1]$ with any other, possibly unbounded, interval Ω.

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 [11,40]. We shall use the common shorthand $D^s f \overset{\text{def}}{=} \frac{\partial^{s_1+\cdots+s_n} f}{\partial^{s_1}\omega_1 \cdots \partial^{s_n}\omega_n}$.

Theorem 2 [14,31] *Suppose $\Omega = [0, 1]$, and consider the functional space*

$$\mathcal{W}_n^r := \left\{ f : \Omega^n \to \mathbb{R}, \max_{\|s\|_\infty \le r} \left\| D^s f \right\|_\infty < \infty \right\}, \tag{7}$$

equipped with the norm $\|f\| = \max_{\|s\|_\infty \le r} \|D^s f\|_\infty$. 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

$$\left| \int_{\Omega^n} f(\omega)\rho(\omega)d\omega - SG_q[f] \right| \le c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} \|f\|. \tag{8}$$

The constant $c_{r,n}$ in Theorem 2 depends only 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 [44]), in general one can not expect to know the exact value of $c_{r,n}$ a priori. Recent results in [17] indicate that $c_{r,n}$ can be exponential in n. This theorem nevertheless provides us with a rate of convergence of the approximation error of sparse grid formulas for numerical integration.

Table 1 compares the number of nodes required by the full grid and the sparse grid formulas to achieve the same 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 10. In contrast, sparse grid formulas using GKP univariate quadrature rules can achieve at least moderate degree of exactness for problems with a few hundred variables. Figure 1 shows Smoljak's sparse grid points using GKP rules for two and three dimensions, for $q = 5$.

We emphasize that sparse grid methods help combat 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

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

Dimension n	Degree d	Full grid (Gaussian)	Sparse grid (GKP)
$n = 5$	$d = 3$	32	11
	$d = 5$	243	71
	$d = 7$	1024	351
	$d = 9$	3125	1471
	$d = 11$	7776	5503
	$d = 13$	16,807	18,943
$n = 10$	$d = 3$	1024	21
	$d = 5$	59,049	241
	$d = 7$	1,048,576	2001
	$d = 9$	9,765,625	13,441
	$d = 11$	6.0466×10^7	77,505
$n = 20$	$d = 3$	1,048,576	41
	$d = 5$	3.4868×10^9	881
	$d = 7$	1.0995×10^{12}	13,201
	$d = 9$	9.5367×10^{13}	154,881
$n = 50$	$d = 3$	1.1259×10^{15}	101
	$d = 5$	7.1790×10^{23}	5201
	$d = 7$	1.2677×10^{30}	182,001
$n = 200$	$d = 3$	1.6070×10^{60}	401
	$d = 5$	2.6561×10^{95}	80,801

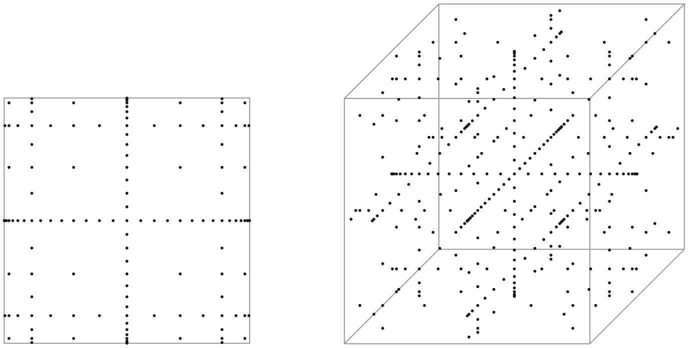

(a) $d = 2$ **(b)** $d = 3$

Fig. 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, **a** $d = 2$, **b** $d = 3$

since for $r \geq 2$ in (8) the term K^{-r} dominates $(\log K)^{(n-1)(r+1)}$. In comparison, MC methods and most classic QMC methods (based on low-discrepancy sequences) do not exploit the differentiability of the integrand [28,29,42]). The rate of convergence for classic QMC methods (including Halton, Sobol, or Niederreiter sequences) is of the order $O(K^{-1}(\log K)^n)$ for sufficiently "smooth" integrands, meaning integrands of bounded Hardy–Krause variation.

Some modern quasi-Monte Carlo methods, in particular *lattice rules* also exploit the differentiability of the integrand as well as additional regularity conditions, such as fast decay in the Fourier coefficients [28, Chap. 5]. See also [21] for the optimal attainable rates of convergence under such assumptions and algorithms to generate lattice rules with optimal rate of convergence in this sense. L'Ecuyer and Munger [22] is a recent software tool for constructing lattice rules with state-of-the-art algorithms.

We shall prove in Sect. 3 that the sparse grid convergence results carry over from integration to stochastic optimization problems. Before that we must consider integration with domains other than $[0, 1]^n$.

2.3 Sparse grid scenario generation for general distributions and domains

The integration domain of the sparse grid method presented in Sect. 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 with the property that if ω is uniformly distributed on $(0, 1)^n$, then $g(\omega)$ is distributed according to P. Then, from Folland [12, 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 can proceed as follows:

1. Choose a $q \geq 1$, and generate the set $\mathcal{G}(q, n) \subset (0, 1)^n$ 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 $\mathcal{G}(q, n)$ to generate the stochastic optimization scenarios.
3. Use the transformed scenarios to approximate the integral $\int_{\Xi} f(\xi)P(d\xi)$.

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

2.4 Sparse grid scenario generation using univariate quadrature rules for general weight functions and affine transformations

It is not essential that the sparse grid formula that we start with in Step 1 above is the sparse grid corresponding to the constant weight function over $(0, 1)^n$; one could analogously transform other sparse grid formulas built using Smoljak's construction from other univariate quadrature formulas supported on any subset of $\mathbb{R}$. See Example 1 below, where the initial sparse grid formula is with respect to the product of univariate normal distributions.

Also 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, $f(g(\cdot))$ may not be a polynomial even when $f(\cdot)$ is. Only when g is affine, is the degree of exactness of the sparse grid formula preserved during the transformation with g. Hence, sparse grid formulas with a prescribed degree of exactness can be computed for every probability measure that is an affine transform of a product of mutually independent probability measures on $\mathbb{R}$. An important special case when such a transformation can be used is 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 [20, Sec.7.4]. The nodes of the Gauss–Hermite quadrature formula of resolution $\nu(\nu = 1, 2, \ldots)$ 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 [13]. The Genz–Keister rule is obtained by a straightforward adaptation of Patterson's algorithm [33] that yields the GKP rule for the uniform distribution. This rule defines a sequence of nested quadrature formulas for the standard normal distribution. 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.

Sparse grid scenarios using Patterson-type nested 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 of polynomial exactness $D_\nu = 2\nu - 1$ with $L(\nu) = \nu$ nodes can be obtained for every continuous distribution using a straightforward algorithm [20, 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 [33] that yields the GKP rule for the uniform distribution and the Genz–Keister rules for the standard normal distribution to obtain nested sequences of quadrature formulas for other continuous distributions with finite moments. One such extension is given in [34]. As mentioned above, it is possible to work without the exact specification of the underlying distribution, and use only the moments of the distribution. The report [24] presents an algorithm that computes nested sequences of univariate quadrature formulas for arbitrary distributions with finite moments directly from the moments of the distribution.

2.5 The degree of exactness of transformed sparse grid formulas

Theorem 3, below, states that for integrands that are polynomials after the diffeomorphism to product form, the approximation (2) using the transformed sparse grid scenarios of Sect. 2.3 gives an exact representation of (1). As mentioned above, the same holds for sparse grid formulas of Sect. 2.4, as the affine transformations map polynomials to polynomials of the same degree, preserving the degree of exactness of the formulas.

Theorem 3 *Consider the optimization problem (1) and its approximation (2), where the weights and scenarios are generated using the sparse grid method, with some control parameter* $q \geq 1$*, using either the approach described in Sect. 2.3 or the approach in Sect. 2.4. 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, therefore the two problems are identical. □

3 The rate of convergence of sparse grid scenario generation

We now give convergence results for the stochastic optimization problems (2) obtained using the approach described in Sect. 2.3.

Theorem 4 presents a 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 [19], Donohue [9], and Pennanen [35] do not apply, and also that the convergence result in Theorem 4 is slightly stronger than the epi-convergence results for the QMC methods in [35].

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

Theorem 4 (Convergence of the sparse grid method) *Consider the optimization problem (1) and its approximation (2), where the scenarios are generated using a transformed sparse grid approximation of Sect. 2.3. Assume that* $\mathcal{C}$ *is closed and bounded, and that for every* $x \in \mathcal{C}$ *the function* $f(\cdot, x)$ *is bounded with* $f(g(\cdot), x) \in \mathcal{W}_n^r$*. Let* x^* *be an optimal solution of (1) with optimal value* z^**, and let* x^K *be an optimal solution of the sparse grid approximation with optimal value* z_K^**. Then*

(i) $z^* \geq \overline{\lim}_K z_K^*$.
(ii) *Every cluster point* $\hat{x}$ *of the sequence* $\{x^K\}_{K=1}^{\infty}$ *is an optimal solution of (1).*

Proof We defer to proof of the first claim, as we give a stronger result, on the rate of convergence of the optimal objective function values, in Theorem 5.

For the second claim, let $F(x) = \int_{\Xi} f(\xi, x)P(d\xi)$, $Q_K(x) = \sum_{k=1}^{K} w^k f(\xi^k, x)$, and let $x^{K_1}, x^{K_2}, \dots$ be a convergent infinite subsequence of $x^1, x^2, \dots$ with limit point $\hat{x}$.

First, observe that because $f(g(\cdot), x) \in \mathcal{W}_n^r$ for every x in the closed and bounded $\mathcal{C}$, there exists a finite constant M satisfying $\|f(g(\cdot), x)\| \le M$ for every x. Now, from Theorem 2 we also have that for every $x \in \mathcal{C}$,

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

implying that the sequence of functions $(Q_{K_t})_{t=1,2,\dots}$ converges to F uniformly on $\mathcal{C}$. Combining this with the continuity of F, we also obtain that for every $\varepsilon > 0$,

$$\left|Q_{K_t}(x^{K_t}) - F(\hat{x})\right| \le \left|Q_{K_t}(x^{K_t}) - F(x^{K_t})\right| + \left|F(x^{K_t}) - F(\hat{x})\right| < \varepsilon$$

for every sufficiently large $K_t > K(\varepsilon)$, that is,

$$\lim_t Q_{K_t}\left(x^{K_t}\right) = F(\hat{x}). \tag{9}$$

We now have the following chain of inequalities:

$$\begin{aligned} \lim_t Q_{K_t}\left(x^{K_t}\right) &\le \lim_t Q_{K_t}(x^*) && (10) \\ &= F(x^*) && (11) \\ &\le F(\hat{x}) && (12) \\ &= \lim_t Q_{K_t}\left(x^{K_t}\right), && (13) \end{aligned}$$

where the inequality (10) follows from the optimality of x^{K_t}, the equation (11) comes from the convergence of the sequence $(Q_{K_t})_{t=1,2,\dots}$ to F, (12) follows from the optimality of x^*, and (13) is the same as (9).

Comparing the left- and right-hand sides of this chain of inequalities we find that all inequalities must hold with equality, therefore $F(\hat{x}) = F(x^*)$, as we claimed. □

Theorem 5 (Rate of convergence) *Consider (1) and its sparse grid approximation (2), where the scenarios are generated using a transformed sparse grid approximation of Sect. 2.3. Assume that $\mathcal{C}$ is closed and bounded, and that for every $x \in \mathcal{C}$ the function $f(\cdot, x)$ is bounded with $f(g(\cdot), x) \in \mathcal{W}_n^r$. Let x^* be an optimal solution of (1), and x^K be an optimal solution of (2), then*

$$\left|\sum_{k=1}^{K} w^k f(\xi^k, x^K) - \int_{\Xi} f(\xi, x^*) P(d\xi)\right| \le \varepsilon, \ \textit{and} \tag{14}$$

$$\left|\int_{\Xi} f(\xi, x^K) P(d\xi) - \int_{\Xi} f(\xi, x^*) P(d\xi)\right| \le 2\varepsilon, \tag{15}$$

where

$$\varepsilon = c_{r,n} K^{-r} (\log K)^{(n-1)(r+1)} \max_{x \in \mathcal{C}} \|f(g(\cdot), x)\|, \tag{16}$$

and $c_{r,n}$, K, $\mathcal{W}_n^r$, and $\|\cdot\|$ are 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 that for every sufficiently large K,

$$F(x^*) - Q_K(x^K) \le F(x^K) - Q_K(x^K) \le \varepsilon,$$

and

$$F(x^*) - Q_K(x^K) \ge F(x^*) - Q_K(x^*) \ge -\varepsilon.$$

This proves (14). Now from (14) and Theorem 2, respectively, we have

$$-\varepsilon \le Q_K\left(x^K\right) - F\left(x^*\right) \le \varepsilon$$
$$-\varepsilon \le F\left(x^K\right) - Q_K\left(x^K\right) \le \varepsilon.$$

Hence, $-2\varepsilon \le F(x^K) - F(x^*) \le 2\varepsilon$. □

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 sparse grid numerical integration.

4 Numerical examples

Our first example (Sect. 4.1) demonstrates the finite convergence of the sparse grid scenario generation for polynomial models. It is a simple example involving the Markowitz model, taken from Rockafellar and Uryasev [39]. The objective, the variance of the return of the portfolio, can be expressed as the integral of a quadratic polynomial.

In Sect. 4.2 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 objective function values to stochastic programs even for high-dimensional problems, regardless of the shape of the underlying distribution.

In the examples we compare variants of the sparse grid method (SG) to MC and QMC sampling. The literature on both of these areas is vast, and it would be unreasonable to attempt a comparison of all the methods that have been proposed in the last few decades, especially as only very few methods have reliable implementations available in the public domain. In our experiments we considered six alternative methods to sparse grids; they belong to three major categories:

1. **MC sampling** As variance reduction is not the concern of this paper, we used a straightforward implementation of MC sampling.
2. **QMC with low-discrepancy sequences** We considered four well-known low-discrepancy sequences: the Halton, reverse-Halton, Sobol, and Niederreiter sequences. Our experiments showed relatively little difference between them, with mostly either the reverse-Halton or the Sobol sequence yielding the best results. As an illustration, we show this similarity in the first example (more numerical results with the other sequences can be found in the technical report [3]); in the remaining problems we only show results with the Sobol sequence.
3. **QMC with optimal lattice rules** "Optimal" lattice rules can be defined using the Fourier or the ANOVA coefficients of the integrands; these parameters are, of course, as hard to compute as numerical integration, and are impossible to even estimate with any reasonable accuracy in stochastic programming, because the integrands change with the decision variables in every iteration of an optimization method. However, it has been reported that defining optimality of lattice rules with respect to suitable surrogate "merit functions" provide lattices that perform well in practical numerical integration [23].

There are still a number of merit functions to choose from and various parameters to tweak to find the optimal lattice parameters with respect to each merit function. The software LatticeBuilder [22] contains various state-of-the-art algorithms for the efficient optimization of rank-one lattice rules with respect to a variety of weight functions and merit functions. We used this software, with settings suggested by its authors in private communication, to generate lattice rules specifically tailored to the smooth problems considered in this section.

For low-dimensional problems ($n < 10$) we used exhaustive search over the lattices' generating vector space to find the best possible lattice. In higher dimensions this is prohibitively expensive, hence we resorted to the fast CBC (component-by-component) algorithm, also implemented in LatticeBuilder, designed for coordinate-symmetric merit functions to find an approximately optimal lattice. This is appropriate since the variables in all of our synthetic problems always play a similar role.

Lattice rules are primarily intended to be used for periodic integrands; for general integrands various periodization strategies can be used to improve performance. We employed the "baker's transformation" (also known as tent transformation), as it is both easily implementable, and can be applied to every integrand.

In our experiments we used the GKP univariate quadrature rules to build sparse grid formulas for the uniform distribution, and analogous Patterson-type nested quadrature rules for non-uniform distributions (recall Sect. 2.4). The code to generate multivariate scenarios from Smoljak sparse grid points was written in Matlab; the approximate

Table 2 Mean returns of assets in Example 2; target return is $R = 0.011$

Instrument	Mean return
S & P	0.0101110
Gov bond	0.0043532
Small cap	0.0137058

Table 3 Covariance matrix of assets in Example 2

	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

problems were solved with an SQP algorithm implemented in Matlab's `fmincon` solver.

After the detailed description of the individual experiments in the next section, the results are summarized in Sect. 4.3.

4.1 Markowitz model

This instance of the classic Markowitz model was used in [37] and [39] for comparing QMC scenario generation methods with the MC method. 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 problem data is given in Tables 2 and 3.

Example 2 (Markowitz model) Let $x = [x_1, \dots, x_n]$ be the amount invested in n 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:

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

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

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

using scenarios x_k and weights w_k.

Table 4 Approximate objective value of the Markowitz model ($\cdot 10^{-3}$)

	Sparse grid		QMC					MC
# Nodes	GKP	Genz–Keister	Sobol	Nieder.	Halton	RevHalton	Lattice	Mean
1	0	0	0	0.115	0.085	0.085	0	2.5546
7	**3**.091	**3.785**	1.099	17.47	1.695	2.253	2.2456	**3**.8511
31	**3**.674		2.990	6.398	2.875	**3**.346	**3**.4612	**3**.3804
111	**3.7**69		**3**.398	4.499	**3**.408	**3**.543	4.0823	**3.8**163
351	**3.78**3		**3**.641	4.023	**3**.640	**3**.683	**3**.8337	**3.77**25
1023	**3.785**		**3.7**26	**3.8**40	**3.7**37	**3.7**37	**3.6**71	**3.8**138
2815	**3.785**		**3.7**60	**3.8**02	**3.7**59	**3.7**67	**3.79**15	**3.79**26

The true optimal objective value is $\approx 3.785 \times 10^{-3}$; bold digits are correct. Both flavors of sparse grid outperform the remaining methods

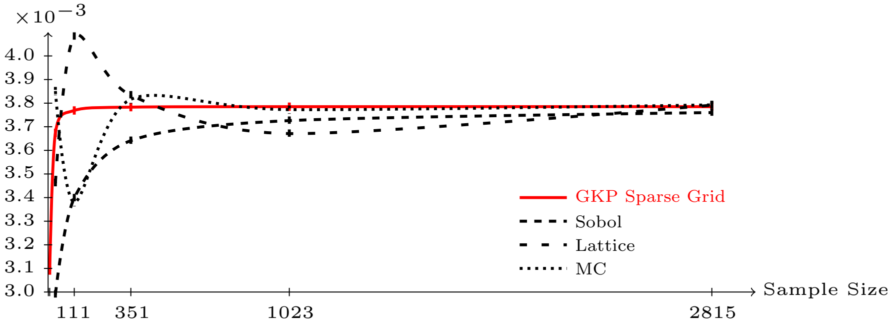


Fig. 2 Approximated objective value of the Markowitz Model. The *y-axis* shows the optimal values from the approximated model for different scenario generation methods for sample size (*x-axis*) from 1 to 2815. The true optimal value is ≈ 0.003785

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 4 and plotted in Fig. 2. (To improve the legibility of the figure, only one of the low-discrepancy sequences is shown, along with MC results and the sparse grid formula that exhibits *slower* convergence, comparing our worst results to MC and QMC.) 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 seven 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. Nevertheless, we obtain 3 correct significant digits with 351 samples, and 4 correct significant digits for sample sizes exceeding 1023; these correspond to $q = 5$ and $q = 6$ in (6). 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, and use the corresponding diffeomorphism. A similar problem arises with QMC methods: one has to ensure (by a random shift, for example) that none of the scenarios lie on the boundary of $[0, 1]^n$. This problem does not arise with sparse grid using the Genz–Keister rule. In general, the boundedness assumption is not needed when the underlying distribution is an affine transformation of a product of univariate distributions, and the sparse grid formula is obtained using an affine transformation g of a sparse grid formula that is built with the univariate densities used as weight functions.

We also generated QMC approximations with the algorithms listed in the beginning of the section; these results are also reported in Table 4. The performance of the low-discrepancy sequences are comparable to each other (except that the Niederreiter sequence is particularly weak with less than 300 points). The optimal lattice rule, obtained using exhaustive search (carried out using LatticeBuilder) to minimize a projection-dependent merit function based on square discrepancy [22], does not yield any improvement over the classic low-discrepancy sequences in this problem.

To confirm the convergence of QMC for this problem, we generated approximations with up to 1,000,000 scenarios using the Sobol sequence and the optimal lattice rule. We observed steady but slow convergence. However, the fourth correct significant digit was obtained only with over 100,000 scenarios in both cases.

It is reasonable to conclude that in this example the sparse grid method achieved much faster convergence than the scenarios generated using popular QMC sequences.

4.2 Utility maximization models

In this section we examine the hypothesis that for sufficiently smooth (but not necessarily polynomial) integrands, sparse grid formulas with high degree of exactness provide a good approximation of the optimal objective function values of stochastic programs even for high-dimensional problems, regardless of the shape of the underlying distribution. For this purpose, we considered utility maximization examples of the form

$$\max_x \int_{\Xi} u\left(x^{\mathrm{T}}\xi\right) p(\xi) d\xi \quad \text{s.t. } \|x\|_1 \leq 1, x \geq 0, \tag{17}$$

for different utility functions u and density functions p.

The three utility functions considered were:

$$u_1(t) = -\exp(t) \qquad \text{(exponential utility),} \tag{18a}$$

$$u_2(t) = \log(1+t) \qquad \text{(logarithmic utility), and} \tag{18b}$$

$$u_3(t) = (1+t)^{1/2} \qquad \text{(power utility).} \tag{18c}$$

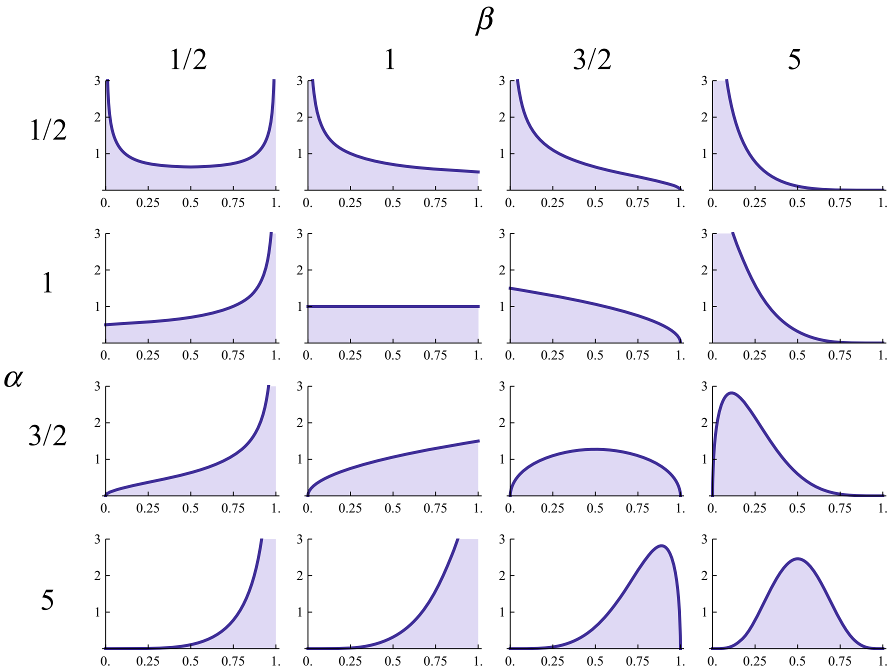


Fig. 3 *Shapes* of Beta (α, β) distributions for $(\alpha, \beta) \in \{1/2, 1, 3/2, 5\}^2$

The probability densities considered were product Beta distributions, obtained by taking the product of univariate Beta(α, β) distributions with $\alpha, \beta \in \{1/2, 1, 3/2, 5\}$ (see Fig. 3. The motivation behind this choice is that it allows us to experiment with distributions of various shapes, and also to transform the problem into product form, for which formulas with different degrees of polynomial exactness can be created and compared. We compared both variants of the sparse grid method: 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 Sect. 2.4 derived for the Beta distribution (or *Patterson-type sparse grid* for short).

4.2.1 Exponential utility

Table 5 shows the (estimated) optimal objective function values computed with different scenario generation techniques for the 100-dimensional exponential utility maximization (using u_1 from (18) in (17)), 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 Fig. 3). In this example the optimal objective function value can be computed relatively easily, because the objective function is the product of one-dimensional integrals; the optimal value is approximately 0.60690986. The table shows great difference in the rate of convergence of the different scenario generation methods. MC integration achieves only 4 correct significant digits using 10^6 scenarios, QMC integration with the Sobol sequence gets 6 digits with about 2×10^5 scenarios, but only 3

Table 5 Results from a 100-dimensional exponential utility maximization example using Beta distributions; comparison of Monte Carlo, two quasi-Monte Carlo methods, and the two Sparse Grid methods

# nodes	MC	QMC (Sobol)	QMC (lattice)	SG (GKP)	Patterson-type SG
201	0.5861458741	**0.605**7992465	**0.60**960463896	**0.6069**012301	**0.606909**7420
20401	**0.6059**951776	**0.6059**86365	**0.6069**3318101	**0.60690986**45	**0.6069098**767
200000	**0.606**784018	**0.606909**7597	**0.60691**252536		
1000000	**0.6069**217022	**0.606909**7569	**0.60691**066152		

The true optimal value is ≈ 0.60690986; bold digits are correct. Both flavors of sparse grid outperform the remaining methods

digits with 2×10^4 scenarios. The optimal rank-one lattice rule achieves 4 correct digits with 2×10^4 scenarios, and one additional digit with 2×10^5 scnarios. In contrast, the transformed GKP rule gets 8 digits with 2×10^4 scenarios. (See column 4 in Table 5.) 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 (see Section 2.4).

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 Fig. 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 Fig. 3 as components. The underlying 160-dimensional product distribution has ten components distributed as each of the distributions shown on Fig. 3. The optimal objective function value is approximately 0.403148407; Table 6 shows the approximate objective function values computed with different techniques using up to half a million scenarios. 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 500,000 scenarios to get the fourth significant digit correctly. Memory constraints prevented the solution of problems with more scenarios.

4.2.2 *Logarithmic utility*

We repeated the above experiments for the logarithmic utility maximization problem, that is, plugging u_2 from (18) into (17), with the same experimental setup. The results were qualitatively very similar to those in the previous section; we only present 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 7.

Table 6 Results from the 160-dimensional exponential utility maximization example using Beta distributions; comparison of Monte Carlo, two quasi-Monte Carlo methods, and the two Sparse Grid methods

# nodes	MC	QMC (Sobol)	QMC (lattice)	SG (GKP)	Patterson-type SG
341	**0.40**20710589	**0.40**27479863	**0.40**507651399	**0.40**07432411	**0.40314**83327
58331	**0.4031**793283	**0.40314**48802	**0.4031**6298189	**0.40**28398967	**0.403148407**1
250000	**0.4031**696956	**0.40314**78092	**0.40315**112035		
500000	**0.4031**910684	**0.403148**3028	**0.40315**062416		

The true optimal value is ≈ 0.403148407; bold digits are correct. Sparse grid built from Patterson-type quadrature formulas outperforms the remaining methods

Table 7 Results from the 160-dimensional logarithmic utility maximization example using Beta distributions; comparison of Monte Carlo, two quasi-Monte Carlo methods, and the two Sparse Grid methods

# nodes	MC	QMC (Sobol)	QMC (lattice)	SG (GKP)	Patterson-type SG
341	−**0.64**97278246	−**0.64**70413764	−**0.64**44548980	−**0.64**93579499	−**0.646451**3847
58331	−**0.646**5609053	−**0.64645**54973	−**0.6464**367282	−**0.646**7992796	−**0.646451**2999
250000	−**0.6464**336612	−**0.646451**9084	−**0.6464**483171		
500000	−**0.6464**842164	−**0.646451**3709	−**0.6464**492263		

The true optimal value is ≈ −0.646451; bold digits are correct. Sparse grid built from Patterson-type quadrature formulas outperforms the remaining methods

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 500,000 nodes for the same accuracy.

4.2.3 Power utility

We repeated the above experiments for the power utility maximization problem, that is, plugging u_3 from (18) into (17), with the same experimental setup. The results obtained with different scenario generation methods are shown in Table 8; they are very similar

Table 8 Results from the 160-dimensional power utility maximization example using Beta distributions; comparison of Monte Carlo, two quasi-Monte Carlo methods, and the two Sparse Grid methods

# nodes	MC	QMC (Sobol)	QMC (lattice)	SG (GKP)	Patterson-type SG
341	−**1.38**63232853	−**1.38**21077423	−**1.38**05140863	−**1.38**35622039	−**1.3816379**583
58331	−**1.381**7632120	−**1.3816**406975	−**1.38163**00171	−**1.38**00861876	−**1.3816379**399
250000	−**1.3816**977241	−**1.38163**82958	−**1.38163**62062		
500000	−**1.38163**63645	−**1.3816379**652	−**1.3816368**776		

The true optimal value is ≈ −1.381638; bold digits are correct. Sparse grid built from Patterson-type quadrature formulas outperforms the remaining methods to the results of the previous experiments. This was the easiest objective for all the methods, but the relative difference between the performance of the different methods remained the same. MC needs over 50,000 scenarios to get to 5 digits of accuracy, the sixth digit is reached with 500,000 scenarios. QMC gets one additional digit of precision with the same effort. Sparse grid with the Patterson-type quadrature rule requires only 341 scenarios for the same accuracy as QMC with 500,000 scenarios.

4.3 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 affinely to a product of univariate distributions, sparse grid scenarios generated using Patterson-type formulas are superior to standard MC and QMC scenario generation methods, including carefully tuned lattice 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 random 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 [24] is also easy, and needs to be carried out only once for every univariate distribution used. The sizes 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 in problems with thousands of random variables the true optimal objective value simply cannot be determined up to reasonable accuracy by alternative methods or statistical bounds, and hence the numerical comparison of results would be meaningless.

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 the high-dimensional examples, although in some instances the results were at least comparable.

5 Concluding remarks

Sparse grid scenario generation appears to be a promising alternative to classic QMC and MC sampling methods, as well as to rank-one lattice rules 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.

Sparse grid formulas using nested quadrature formulas have fewer 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 of these 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 the uniform distribution. Scenarios with a given degree of polynomial exactness are easily generated for distributions that are affine transformations of a product of univariate distributions—this includes multivariate normal distributions. We were able to generate univariate quadrature formulas of at least 5 different resolutions (with degrees of exactness exceeding 40) for all the distributions considered in this paper. The limits of this approach is an open problem; for example it is not known for what weight functions are there GKP formulas for every resolution [26]. Note that this is primarily a theoretical gap with no practical consequence: first, because high-dimensional sparse grids of high resolution have prohibitively large number of nodes, so one only needs low-resolution univariate formulas, and second, because Gaussian formulas with arbitrarily high degree of polynomial exactness are available for every univariate distribution, and these can be used even in the absence of Patterson-type formulas.

It is important to note that to obtain the appropriate univariate quadrature formulas and the sparse grid scenarios, one does not need exact distributional information, it is sufficient to know the moments of the underlying probability distribution up to the same degree as the degree of exactness of the sought sparse grid formula.

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: $O(n)$ scenarios give polynomial exactness of degree 3 for n random variables; $O(n^2)$ scenarios provide exactness of degree 5, and $O(n^3)$ scenarios provide degree 7 polynomial exactness. The only limiting factors in the sizes of the problems considered in the numerical experiments section were the size of the memory of the computer used for these experiments, and the fact that none of the alternative methods provided fast enough convergence to the optimum that the Patterson-type sparse grid seemed to have converged to—so our results for the higher-dimensional problems simply could not be validated using alternative methods. Upon submission of the manuscript, an anonymous referee pointed us to some very recent theoretical results on the intractability of the space $\mathcal{W}_n^r$ for numerical integration [17]. The stark contrast between theory and practice is perhaps worthy of further investigation.

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 might make the approximation of the convex problems non-convex. While this did not result in any problems in the experiments presented in the paper, a theoretically satisfying resolution of this issue is an important topic of future research.

Acknowledgments Research supported in part by Grants DOE DE-FG02-10ER26037, SC0005102; DOE-SP0011568; and ONR N00014210051. We thank David Munger for the discussion on optimal lattice rules for smooth integrands and the help with his LatticeBuilder software. We also thank the associate editor and the referees for the constructive suggestions that helped improve the manuscript.

References

1. Bungartz, H.J., Griebel, M.: Sparse grids. Acta Numer. **13**, 147–269 (2004). doi:10.1017/S0962492904000182
2. Casey, M., Sen, S.: The scenario generation algorithm for multistage stochastic linear programming. Math. Oper. Res. **30**, 615–631 (2005)
3. Chen, M., Mehrotra, S.: Epi-convergent scenario generation method for stochastic problems via sparse grid. Tech. rep., Northwestern University (2007). http://edoc.hu-berlin.de/series/speps/2008-7/PDF/7.pdf
4. Consigli, G., Dupačová, J., Wallace, S.: Generating scenarios for multistage stochastic programs. Ann. Oper. Res. **100**, 25–53 (2000)
5. Davis, P.J., Rabinowitz, P.: Methods of Numerical Integration. Academic Press, San Diego (1975)
6. Dempster, M.A.H., Thompson, R.T.: EVPI-based importance sampling solution procedure for multistage stochastic linear programming on parallel MIMD architectures. Ann. Oper. Res. **90**, 161–184 (1999)
7. Dick, J., Pillichshammer, F.: Digital Nets and Sequences. Cambridge University Press, Cambridge (2010)
8. Donohue, C.: Stochastic network programming and the dynamic vehicle allocation problem. Ph.D. thesis, The University of Michigan, Ann Arbor (1996)
9. Donohue, C.J., Birge, J.R.: The abridged nested decomposition method for multistage stochastic linear programs with relatively complete recourse. Algorithmic Oper.Res. **1**, 20–30 (2006)
10. Dupačová, J., Gröwe-Kuska, N., Römisch, W.: Scenario reduction in stochastic programming: an approach using probability metrics. Math. Program. **95**, 493–511 (2003)
11. Evans, L.C.: Partial Differential Equations. American Mathematical Society, Providence (1998)
12. Folland, G.B.: Real Analysis: Modern Techniques and Their Applications, 2nd edn. Wiley, Hoboken (1999)
13. Genz, A., Keister, B.: Fully symmetric interpolatory rules for multiple integrals over infinite regions with gaussian weight. J. Comput. Appl. Math. **71**(2), 299–309 (1996). doi:10.1016/0377-0427(95)00232-4
14. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. Numer. Algorithms **18**, 209–232 (1998)
15. Heiss, F., Winschel, V.: Likelihood approximation by numerical integration on sparse grids. J. Econom. **144**(1), 62–80 (2008). doi:10.1016/j.jeconom.2007.12.004
16. Heitsch, H., Römisch, W.: Scenario tree modeling for multistage stochastic programs. Math. Program. **118**(2), 371–406 (2007). doi:10.1007/s10107-007-0197-2
17. Hinrichs, A., Novak, E., Ullrich, M., Wozniakowski, H.: The curse of dimensionality for numerical integration of smooth functions. Math. Comput. **83**, 2853–2863 (2014)
18. Kaut, M., Wallace, S.W.: Evaluation of scenario-generation methods for stochastic programming. Pac. J. Optim. **2**, 257–271 (2007)
19. King, A.J., Wets, R.J.B.: Epi-consistency of convex stochastic programs. Stoch. Stoch. Rep. **34**, 83–92 (1991)
20. Krylov, V.I.: Approximate Calculation of Integrals. Dover, Mineola (2005)
21. Kuo, F.: Component-by-component constructions achieve the optimal rate of convergence for multivariate integration in weighted Korobov and Sobolev spaces. J. Complex. **19**(3), 301–

320 (2003). doi:10.1016/S0885-064X(03)00006-2. http://www.sciencedirect.com/science/article/pii/S0885064X03000062

22. L'Ecuyer, P., Munger, D.: LatticeBuilder: A general software tool for constructing rank-1 lattice rules. Submitted (2012). Software downloadable from http://www.iro.umontreal.ca/simardr/latbuilder/latbuilder.html
23. L'Ecuyer, P., Munger, D.: On figures of merit for randomly-shifted lattice rules. In: Wozniakowski, H., Plaskota, L. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2010, pp. 133–159. Springer, Berlin (2012)
24. Mehrotra, S., Papp, D.: Generating nested quadrature formulas for general weight functions with known moments. Tech. Rep. arXiv:1203.1554v1 [math.NA], Northwestern University (2012). http://arxiv.org/abs/1203.1554v1
25. Mehrotra, S., Papp, D.: Generating moment matching scenarios using optimization techniques. SIAM J. Optim. **23**(2), 963–999 (2013). doi:10.1137/110858082
26. Monegato, G.: Stieltjes polynomials and related quadrature rules. SIAM Rev. **24**(2), 137–158 (1982)
27. Neumaier, A.: Introduction to Numerical Analysis. Cambridge University Press, Cambridge (2001)
28. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 63. SIAM, Philadelphia (1992)
29. Niederreiter, H., Talay, D. (eds.): Monte Carlo and Quasi-Monte Carlo Methods. Springer, Berlin (2006)
30. Norkin, V., Pflug, G., Ruszczyński, A.: A branch and bound method for stochastic global optimization. Math. Program. **83**, 425–450 (1998). doi:10.1007/BF02680569
31. Novak, E., Ritter, K.: High dimensional integration of smooth functions over cubes. Numer. Math. **75**, 79–97 (1996)
32. Novak, E., Ritter, K.: Simple cubature formulas with high polynomial exactness. Constr. Approx. **15**, 499–522 (1999)
33. Patterson, T.N.L.: The optimal addition of points to quadrature formulae. Math. Comput. **22**(104), 847–856 (1968)
34. Patterson, T.N.L.: An algorithm for generating interpolatory quadrature rules of the highest degree of precision with preassigned nodes for general weight functions. ACM Trans. Math. Softw. **15**, 123–136 (1989). doi:10.1145/63522.63523
35. Pennanen, T.: Epi-convergent discretizations of multistage stochastic programs. Math. Oper. Res. **30**, 245–256 (2005)
36. Pennanen, T., Koivu, M.: Integration quadrature in discretization of stochastic programs. Stoch. Program. E-Print Series (2002–11)
37. Pennanen, T., Koivu, M.: Epi-convergent discretizations of stochastic programs via integration quadratures. Numer. Math. **100**, 141–163 (2005). doi:10.1007/s00211-004-0571-4
38. Pflug, G.C.: Scenario tree generation for multiperiod financial optimization by optimal discretization. Math. Program. **89**, 251–271 (2001)
39. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. J. Risk **2**(3), 21–41 (2000)
40. Rudin, W.: Real and Complex Analysis. McGraw-Hill, New York (1986)
41. Smoljak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Dokl. Akad. Nauk SSSR **4**, 240–243 (1963)
42. Sobol, I.M.: A Primer for the Monte Carlo Method. CRC Press, Boca Raton (1994)
43. Wallace, S.W., Ziemba, W.T. (eds.): Applications Of Stochastic Programming. Society for Industrial and Applied Mathematics, Philadelphia (2005)
44. Wasilkowski, G.W., Wozniakowski, H.: Explicit cost bounds of algorithms for multivariate tensor product problems. J. Complex. **11**(1), 1–56 (1995)