Epi-convergent Scenario Generation Method for Stochastic Problems via Sparse Grid [*]

Michael Chen[†] Sanjay Mehrotra[‡]

December 14, 2007

Technical Report 2007-08
Department of Industrial Engineering and Management Sciences,
Robert R. McCormick School of Engineering,
Northwestern University, Evanston, Illinois 60208.

Abstract

One central problem in solving stochastic programming problems is to generate moderate-sized scenario trees which represent well the risk faced by a decision maker. In this paper we propose an efficient scenario generation method based on sparse grid, and prove it is epi-convergent. Furthermore, we show numerically that the proposed method converges to the true optimal value fast in comparison with Monte Carlo and Quasi Monte Carlo methods.

1 Introduction

Multistage stochastic programming models many important sequential decision making problems, such as optimal forest harvest, portfolio optimization, pension fund management, optimal routing, option pricing, asset liability management, optimal consumption etc, see Wallace and Ziemba [39]. A T-stage sequential decision making process under constraints is as following. Let $\{\xi_t\}_{t=0}^T, T < \infty$ be a finite horizon stochastic process, where $\xi_t \in \Xi_t \subsetneqq \mathbb{R}^{d_t}$, Ξ_t be Borel sets $\forall t = 0, \ldots, T$. We let Ξ_0 be a singleton, which means that the current data is known. Let

$$\xi = [\xi_0, \xi_1, \ldots, \xi_T],\ \ \Xi = [\Xi_0, \Xi_1, \ldots, \Xi_T]\,. \tag{1}$$

The space Ξ is supplied with a probability distribution for ξ, i.e., ξ is properly defined in the space $(\Xi, \mathcal{F}, P)$, where $\mathcal{F}$ is the Borel $\sigma-$field on Ξ and P is the probability measure on $(\Xi, \mathcal{F})$. We assume that the probability distribution could be represented by a density function. We allow the individual $\xi_t, t = 0, \ldots, T$ depends on each other within stage or across the stages. However, we assume that the probability structure is unchanged by our decisions.

[*] Supported by NSF grants DMI-0200151, DMI-0522765 and ONR grant N00014-01-1-0048.

[†] Department of IE/MS, Northwestern University, Evanston, IL 60208, m-chen@northwestern.edu

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

We define a *policy* as $x(\cdot) : \Xi \to \mathbb{R}^d, d = d_0 + d_1 + \ldots + d_T$. We emphasize that $x(\cdot)$ has the following nonanticipative structure:

$$x(\xi) = [x_0(\xi_0), \ldots, x_T(\xi_0, \ldots, \xi_T)]\,,$$

where each $x_t(\cdot) : \Xi_0 \times \ldots \times \Xi_t \to \mathbb{R}^{d_t}$. We let the nonanticipative function space be $\mathcal{N}$. The evolution of available information and the corresponding adaptive decisions are the key gradients in this sequential decision problem. Though the focus is on the current stage decision, it can not be properly optimized without taking into account the risk and the optimal recourse action in the future. By deciding a solution $x_t(\xi_0, \ldots, \xi_t), \forall t = 1, \ldots, T$ at the current stage, we choose in advance exactly what to do for all possible outcomes.

In addition to be nonanticipative, each $x_t(\cdot)$ is also subject to constraints given by inequalities

$$C_t(\xi_{\mathbf{t-1}}, x_{\mathbf{t-1}}) = \{x_t(\cdot) \,|\, g_t(\xi_{\mathbf{t-1}}, \xi_t, x_{\mathbf{t-1}}, x_t(\cdot)) \geq 0 \text{ P-a.s.}\}\,,$$

where we use the multi-index $\mathbf{t} = [0, 1, \ldots, t]$ and let $\xi_{\mathbf{t-1}} = [\xi_0, \ldots, \xi_{t-1}]$ and $x_{\mathbf{t-1}} = [x_0, \ldots, x_{t-1}]$. $g_t(\cdot) \geq 0$ means a set of inequality constraints of stage t.

The cost associated with outcomes and our decisions is given in function $f : \Xi \to \mathbb{R}$. We assume f to be continuous for every $\xi \in \Xi$ and measurable on $(\Xi, \mathcal{F}, P)$. The goal of a decision maker is to minimize the expected cost by choosing an optimal policy $x(\cdot)$ subject to constraints, which is expressed in a compact form as:

$$\begin{aligned} &\text{minimize} \int_{\Xi} f(\xi, x(\xi)) d_{P(\xi)} \\ &\text{subject to } x(\cdot) \in \mathcal{N}. \cap C \end{aligned} \qquad (\mathcal{P})$$

We can also formulate ($\mathcal{P}$) by decomposition into stages, which views the recourse stage *parametrically*, i.e., $\rho_t(\cdot, x_{t-1}) : \Xi_t \to \mathbb{R}, \forall t = 1, \ldots, T$.

$$\begin{aligned} \rho(\xi_0) =& \min f_0(\xi_0, x_0) \star \int_{\Xi_1} \rho_1(\xi_1, x_0) dP(\xi_1), \\ & \text{where } x_0 \in C_0(\xi_0), \\ \rho_{t-1}(\xi_{t-1}, x_{t-1}) =& \min f_{t-1}(\xi_{t-1}, x_{t-1}) \star \int_{\Xi_{t-1}} \rho_t(\xi_t, x_{t-1}) dP(\xi_t), \\ & x_{t-1} \in C_{t-1}(\xi_{t-1}, x_{t-2}) \;\; \forall t = 2, \ldots, T-1, \\ \rho_T(\xi_T, x_{T-1}) =& \min f_T(\xi_T, x_T), \\ & x_T \in C_T(\xi_T, x_{T-1}). \end{aligned}$$

The $\star$ notation emphasizes that the costs between stages could be additive, multiplicative, or take another form. For example, for the most widely used stochastic linear models the cost takes the additive form across stages:

$$f(\xi) = f_0(\xi_0, x_0(\xi_0)) + \ldots + f_T(\xi_T, x_T(\xi_{\mathbf{T}})). \qquad (2)$$

While for many other problems, for example, the Utility Maximization model in Section 5, the relation of costs between stages is multiplicative,

$$f(\xi) = f_0(\xi_0, x_0(\xi_0)) \cdot f_1(\xi_1, x_1(\xi_\mathbf{1})) \ldots f_T(\xi_T, x_T(\xi_\mathbf{T})). \tag{3}$$

In general, we assume that the $\star$ operator preserves the boundness of its two operands.

The recourse decomposition representation emphasizes that the cost $\rho_t(\cdot) : \Xi_t \to \mathbb{R}$ is a function of ξ_t alone, where the optimal solution x_t and future optimal policy $x(\xi_{t+1}), \ldots, x(\xi_T)$ are assumed. To emphasize the connection between stages in this recursive definition, we write $\rho_t(\xi_t, x_{t-1})$ instead of $\rho_t(\xi_t)$.

2 Approximating integration using Scenarios

Due to high dimensionality it is usually impossible to evaluate the integral of the objective function $f(\xi, x(\xi))$ with a given candidate policy $x(\xi)$. In practice this high dimensional integral over continuous distribution is approximated linearly, i.e., we generate $K < \infty$ scenarios $\{\xi^k\}_{k=1}^K$, where each $\xi^k = [\xi_1^k, \xi_2^k, \ldots, \xi_T^k] \in \mathbb{R}^N$, and assign weights w_k to scenarios. Various scenario generation methods have been proposed to determine the scenarios and weights. Among them the simplest Monte Carlo method uses pseudo random number (vectors) as scenarios and $\frac{1}{K}$ as weights for all scenarios. Pennanen and Koivu [30] use low-discrepancy sequences (Quasi Monte Carlo methods), such as Faure sequence, Sobol sequence, and Niederreiter sequence. The Quasi Monte Carlo methods also use $\frac{1}{K}$ as weights for all scenarios. Methods by Dupačová, Gröwe-Kuska and Römisch [13], Heitsch and Römisch [17], Pflug [33], G. Consigli and J. Dupačová and S. Wallace [8], Michael and Wallace [24], and Pennanen [30] focus mainly on the properties of the stochastic process $\{\xi_t\}_{t=1}^T$. Dupačová, Gröwe-Kuska and Römisch [13] and Pflug [33] reduce the approximation error by minimizing the probability metrics. Statistical techniques like cluster analysis, importance sampling and moment matching have been applied in the scenario generation by Consigli, Dupačová and Wallace [8] and Høyland, Kaut and Wallace [24]. The methods by Dempster and Thompson [10], and Casey and Sen [5] take into account not only the stochastic process $\{\xi_t\}_{t=1}^T$ but also the properties of the integrand function $f(\xi, x(\xi))$ in generating the scenario tree. Dempster and Thompson [10] determine the number of branches dynamically based on an estimate of the expected value of perfect information. Casey and Sen [5] apply sensitivity analysis of linear programming to guide the process of scenario generation. All of the above mentioned methods use positive weights for scenarios they generate. Interestingly, the sparse grid method for generating scenarios proposed in this paper generates both positive and negative weights.

We approximate the $(\mathcal{P})$ as:

$$\begin{aligned} &\text{minimize} \sum_{k=1}^K w_k f(\xi_k, x(\xi_k)) \\ &\text{subject to } x \in \mathcal{N} \cap C. \end{aligned} \tag{$\mathcal{P}^K$}$$

We assume relative completeness for both $(\mathcal{P})$ and $(\mathcal{P}^K)$. One could use artificial variables with penalty to achieve relative completeness of $(\mathcal{P})$. In rare case, even if $(\mathcal{P})$ has relative completeness, an individual scenario with measure zero might be picked up and cause infeasibility of $(\mathcal{P}^K)$. Hence, the relative completeness of $(\mathcal{P}^K)$ for any K scenarios is required to justify the approximation. For two-stage linear problem, the fixed recourse and the finite second moment of ξ are sufficient conditions of the assumption, see Birge and Louveaux [3] Theorem 3.

Let z^* and z^*_K be the optimal objective values of $(\mathcal{P})$ and its approximation $(\mathcal{P}^K)$ respectively. The quality of a scenario generation procedure depends on whether or not the optimal value z^*_K of the approximated problem converges to the true optimal value z^*. And furthermore, if the true optimal solution is unique, the quality of a discretization procedure also depends on whether or not the sequence of approximated solutions x^*_K as $K \to \infty$ converges to the optimal solution x^*. In case where both the original minimization problem and the approximated problems might have many optimal solutions, we only expect a cluster point (if any) of a sequence $\{x^*_K\}_{K=1}^{\infty}$ be an optimal solution x^*. The problem is how to generate a set of scenarios $\{\xi^k\}_{k=1}^{K}$ and the corresponding weights w_k such that the approximation error $|z^* - z^*_K|$, is small and the scenario generation method is *consistent*, i.e.,

$$\lim_{K\to\infty} |z^* - z^*_K| = 0. \tag{4}$$

A concept for analyzing asymptotic behavior of a discretization procedure is *epi-convergence*, as proposed by Rockafellar and Wets [35]. Epi-convergence of Monte Carlo sampling-based methods has been studied King and Wets [21] and Donohue [11]. Epi-convergence of low-discrepancy sequences based method is established by Pennanen and Koivu [31], and Pennanen [28], [29].

Since the epi-convergence results are asymptotic, for comparison purposes we usually test different discretization procedures over stochastic programs with known optimal objective values. In our numerical examples in section 5, we used Markowitz model and CVaR optimization. Pennanen and Koivu [31] tested quasi-sequence based discretization procedure on Markowitz model, Kaut and Wallace [20] used CVaR optimal portfolio model. A more general approach, even when the true optimal value is unknown, is to derive the *statistical* upper and lower bound by Monte Carlo method as shown in Chiralaksanakul and Morton [7], and to compare different discretization procedures against those bounds. Pennanen and Hilli [32] support their epi-convergent conclusions by numerical examples of multistage stochastic optimization problems, and show the convergence of their method by comparing the results with the statistical bounds. Kaut and Wallace [19] give a comprehensive overview and practical guidelines for evaluating discretization procedures for stochastic program, which states that "We are not concerned about how well the distribution is approximated, as long as the scenario tree leads to a "good" decision."

We propose an effective scenario generation method based on sparse grid for problems with continuous distributions in Section 3. We prove epi-convergence of the new discretization method in Section 3. In Section 5 we show numerically that the convergence rate of proposed sparse grid method is favorable in comparison with the Quasi Monte Carlo. In Section 5.1 we demonstrate the finite convergence property of the sparse grid method, in Section 5.2 we demonstrate rapid convergence of this method for smooth functions, and in Section 5.3 we discuss the performance of this method on piece-wise linear functions. For the purpose of these comparisons we use Markovitz, Maximum Utility, and CVaR portfolio optimization models arising in finance. We also give an effective dual transformation for solving the two-stage linear problems based on the extensive formulation in Appendix 5.4.

3 Sparse Grid scenario generation algorithm

For one dimensional integration univariate quadrature algorithms are known to be very effective. Examples of univariate quadrature algorithms are Newton-Cotes (midpoint, rectangle, trapezoidal), Gauss quadratures (Legendre, Chebyshev, Laguerre, Hermite, Jacobi, Kronrod, Patterson), and Clenshaw-Curtis rule, see Davis and Rabinowita [9], Neumaier [25]. For a given univariate quadrature rule the number of scenarios is specified by a function $L(\nu) : \mathbb{N} \to \mathbb{N}$, where ν is called resolution. For example, in the easiest midpoint rule $L(\nu) = 2^{\nu} - 1$. With ν given, a univariate quadrature algorithm generates a set of scenarios and corresponding weights and approximate an integral linearly:

$$\int_{[0,1]} f(\xi) d_{\xi} \approxeq \sum_{k=1}^{L(\nu)} w_k f(\xi_k). \tag{Q^{ν}}$$

One important feature of a univariate quadrature algorithm is its *degree of polynomial exactness*. When f is a polynomial function, the approximation is exact for a sufficient large ν. We use $P(\nu) : \mathbb{N} \to \mathbb{N}$ to denote the highest polynomial degree for which the approximation is exact. In general, different univariate quadrature algorithms have different $L(\nu)$, output different scenarios and weights, and have different degree of polynomial exactness $P(\nu)$. For a thorough treatment of these univariate quadrature rules we refer readers to Davis and Rabinowita [9], Neumaier [25], and Bungartz and Griebel [4].

However it is inefficient to extend the univariate quadrature algorithms to the high dimensional quadrature algorithm *directly*, i.e., applying the univariate quadrature algorithm to each dimension. This so called "full grid" algorithm will inevitably suffer from *curse of dimensionality*. The "sparse grid" algorithm by Smolyak [37], utilizes any of the above univariate quadrature algorithms and uses much less scenarios than a "full grid" algorithm does. Here we briefly describe Smolyak's construction of multivariate quadrature and summarize major results.

3.1 Smolyak's sparse grid algorithm

Given a univariate quadrature formula Q^{ν}, let

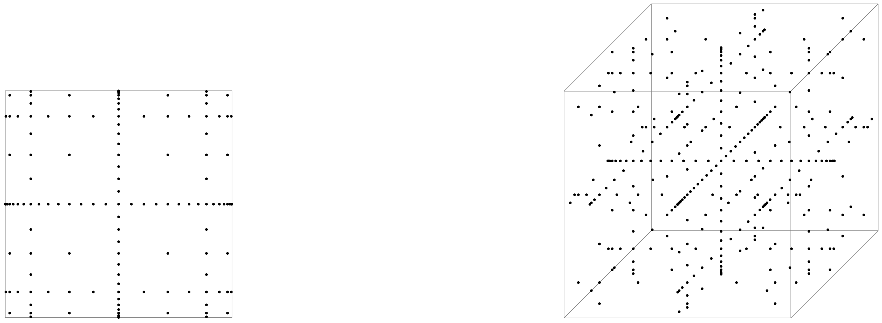
$$\Xi_{\nu} = \{\xi_1^{\nu}, \dots, \xi_{L_{\nu}}^{\nu}\}, w_{\nu} = \{w_1, \dots, w_{L_{\nu}}\}. \tag{5}$$

Smolyak's d-dimensional quadrature algorithm restricts the resolution along each of the d dimension, i.e., $\nu_1, \cdots, \nu_d$ into a simplex. Or equivalently, let $\nu = [\nu_1, \cdots, \nu_d]$, then Smolyak's algorithm requires $q \le \|\nu\|_1 \le q + d - 1$ with some specified parameter $q \in \mathbb{N}$. Hence, the set of scenarios generated by Smolyak's algorithm is

$$\mathcal{H}(q, m) = \bigcup_{q \le \|\nu\|_1 \le q+m-1} (\Xi_{\nu_1} \times \cdots \times \Xi_{\nu_m}). \tag{6}$$

By varying the parameter q, we decide how many points to use in the approximation in ($\mathcal{P}_{\nu}$) . If $\Xi_{\nu} \cap \Xi_{\nu+1} = \emptyset$, then the number of scenarios are

$$\sum_{q \le \|\nu\|_1 \le q+m-1} L_{\nu_1} \cdots L_{\nu_m}. \tag{7}$$



(a) $d = 2$

(b) $d = 3$

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

The weights are also combination of $w_{\nu_1}, w_{\nu_2}, \dots, w_{\nu_d}$, and by Wasilkowski and Wozniakowski [41] the Smolyak's algorithm approximate the multivariate integral $\int_{[0,1]^d} f(\xi) d_\xi$ by

$$\sum_{q\le\|\nu\|_1\le q+d-1} (-1)^{q+d-\|\nu\|_1-1} \left(\begin{array}{c} d-1 \\ \|\nu\|_1-1 \end{array}\right) \sum_{k_1=1}^{L_{\nu_1}} \cdots \sum_{k_d=1}^{L_{\nu_d}} w_{k_1} \dots w_{k_d} f(\xi_{k_1}, \dots, \xi_{k_d}). \tag{8}$$

If $\Xi_\nu \subset \Xi_{\nu+1}$, then the univariate quadrature rule is *nested*, and hence

$$\mathcal{H}(q,d) \subset \mathcal{H}(q+1,d). \tag{9}$$

With nested quadrature rules $\mathcal{H}(q,d)$ has much less cardinality. To see this, let us consider the one-dimensional difference grids

$$\tilde{\Xi}_\nu = \Xi_\nu \backslash \Xi_{\nu-1}, \tag{10}$$

where $\Xi_0 = \emptyset$. We denote cardinality of $\tilde{\Xi}_\nu$ as $\tilde{L}_\nu$. Hence, the points of the Smolyak's multivariate formulas are

$$\tilde{\mathcal{H}}(q,d) = \bigcup_{q\le\|\nu\|_1\le q+m-1} (\tilde{\Xi}_{\nu_1} \times \cdots \times \tilde{\Xi}_{\nu_m}), \tag{11}$$

and the number of points in the set $\tilde{\mathcal{H}}(q,m)$ is

$$\sum_{q\le\|\nu\|_1\le q+d-1} \tilde{L}_{\tilde{\nu}_1} \dots \tilde{L}_{\tilde{\nu}_m}. \tag{12}$$

For example, if $L_\nu = 2^{\nu-1}+1, \nu \ge 2$ and $L_1 = 1$, then $\tilde{L}_\nu = 2^{\nu-2}, \nu \ge 3$ and $\tilde{L}_\nu = \nu, \nu = 1, 2$. Then for $d = 2, q = 9$, number of sparse grids is 1537, while for a non-nested univariate quadrature with the same L_ν function, number of points in the corresponding multivariate quadrature rule is 4105. In terms of two-stage stochastic programming, we would need to solve extra 2568 subproblems should we use non-nested univariate quadrature rules as input of Smolyak's construction. Figure 1 shows Smolyak's sparse grids using Gauss-Patterson univariate quadrature algorithm for two and three dimensions using $q = 5$.

We now introduce the error estimate of sparse grid method in approximating high dimensional integration. First of all, it is shown by [2] that Smolyak's multivariate quadrature preserves the underlying univariate quadrature polynomial exactness.

Corollary 3.1 *(Barthemann, Novak and Ritter[2]) Sparse grid approximation (8) is exact for polynomials in the space $\sum_{q\le|\nu|_1\le q+d-1}(\mathcal{P}_{\nu_1}\otimes\cdots\otimes\mathcal{P}_{\nu_d})$, where $\mathcal{P}_{\nu_i}$ means univariate polynomial space with degree $P(\nu_i)$.*

More generally, error estimate of Smolyak's sparse grid algorithm exists for the following function space.

$$\mathcal{W}_d^r := \left\{ f : [0,1]^d \to \mathbb{R}, \left\| \frac{\partial^{s_1+s_2+\ldots+s_d} f}{\partial^{s_1}_{\xi_1}\ldots\partial^{s_d}_{\xi_d}} \right\|_\infty < \infty, \max(s_1,s_2,\ldots,s_d) \le r \right\}. \tag{13}$$

Note that the space is defined through *weak derivative*, instead of the usual partial derivative, see Walter Rudin [36]. Especially d-dimensional piecewise linear functions belong to $\mathcal{W}_r^1$.

The following theorem is well known.

Theorem 3.1 *(Smolyak[37]) For $d, r \in \mathbb{N}$, $f \in \mathcal{W}_d^r$, let K be the cardinality of the sparse grid $\mathcal{H}(d,q)$, then there exists $c_{r,m} > 0$ such that*

$$\sup_{f\in\mathcal{W}_d^r} \left| \int_{[0,1]^d} f(\xi) d_\xi - \sum_{k=1}^K w_k f(\xi_k) \right| \le c_{r,d} K^{-r} (\log K)^{(d-1)(r+1)} \|f\|_\infty. \tag{14}$$

Note that $c_{r,d}$ is a constant depending on d, r and the underlying univariate quadrature rule used by the sparse grid method. For some cases, $c_{r,d}$ could be calculated explicitly as well e.g., see Wasilkowski and Woźniakowski [41]. We emphasize that sparse grid method overcomes the "curse of dimensionality" up to a logarithmic factor in the term $(\log K)^{(d-1)(r+1)}$. For a given problem of dimension d, the error goes to zero fast since the term K^{-r} will dominate $(\log K)^{(d-1)(r+1)}$. For more recent development in this area see Bungartz and Griebel [4].

3.2 Transformation from the Constructive Dimension

In the classical sparse grid method the integration domain is the unit cube $[0,1]^d$ and the random variable is uniformly distributed. For applications with different random variable and integration domain we need to have a change of variable before applying the sparse grid method. Suppose the random variable $\xi \in \mathbb{R}^n$ is generated from a uniform random variable $\omega \in \mathbb{R}^n$, i.e. $\xi = g(\omega)$, and g is a diffeomorphism, then we have

$$\int_{g([0,1]^d)} f(\xi) d_{F(\xi)} = \int_{[0,1]^d} (f\circ g)(\omega) d_\omega. \tag{15}$$

For g to be a diffeomorphism, g has to be continuously differentiable and a bijection. We refer readers to Folland [14, Theorem 2.47] for the definition and the rule of change of variable for Lebesgue integration. We show the change of variable in the following two important models, each of them covers a wide range of applications.

Model 3.1 *(multivariate one-stage process) Let $\xi = \mu + \sigma\epsilon$, where $\mu \in \mathbb{R}^d$, $\sigma \in \mathbb{R}^{d\times d}$ are constants, and ϵ is a standard d-dimensional normal random vector. Let ϕ be the cumulative distribution function of univariate normal random variable. If σ is invertible, then g is a diffeomorphism, and*

$$\mathbf{g}(\omega) = \mu + \sigma[\phi^{-1}(\omega_1), \cdots, \phi^{-1}(\omega_d)]^T. \tag{16}$$

As shown g is naturally defined by the definition of the random variable ξ. For a T-stage process such g is also intrinsically implied by the model of the random process $\{\xi_t\}_{t=0}^T$ as shown by the following model.

Model 3.2 *(multivariate multistage process) Let* $\xi = [\xi_0, \dots, \xi_T]$, $\omega = [\omega_1, \dots, \omega_T]$, $\xi_t \in \mathbb{R}^{d_t}$, $\omega_t \in \mathbb{R}^{d_t}$, $t = 1, \dots, T$. *Let* $\xi_t = \mu_t(\xi_0, \dots, \xi_{t-1}) + \sigma_t(\xi_0, \dots, \xi_{t-1})\epsilon_t$, *where* $\mu_t : \mathbb{R}^{d_0+\dots+d_{t-1}} \to \mathbb{R}^{d_t}$, $\sigma_t : \mathbb{R}^{d_0+\dots+d_{t-1}} \to \mathbb{R}^{d_t \times d_t}$ *, and* ϵ_t *are* d_t *dimensional standard normal random vector* $\forall t = 1, \dots, T$. *Then* $g(\omega) = [\xi_1, \dots, \xi_T]$ *is a diffeomorphism if all* $\sigma_t, \forall t = 1, \dots, T$ *are invertible.*

This model covers a wide range of financial time series models, such as Vector Auto Regressive (VAR), Vector Equilibrium Correction (VEqC), Vector Autoregressive Moving Average (VARMA), Auto Regressive Conditionally Heteroscedastic (ARCH), and its generalized model GARCH. The method of generating random variable through inverse transformation is not limited to normal random variable. It applies to a wide range of distributions and we refer readers to Law and Kelton [22] for more detailed introduction of the subject.

3.3 Sparse Grid scenario generation algorithm

We now propose the following scenario generation algorithm for a T-stage problem with random process $\xi = [\xi_1, \dots, \xi_T], \xi_t \in \mathbb{R}^{d_t}$, $d = d_1 + \dots + d_T$. We require that $\xi = g(\omega)$, and g is a diffeomorphism, $g : \mathbb{R}^d \to \mathbb{R}^d$.

1. Determine an univariate quadrature rule and q;
2. Generate the set $\mathcal{H}(q, d) \subset [0, 1]^d$ of K scenarios and the corresponding weights by the standard sparse grid algorithm.
3. Apply g transformation to $\mathcal{H}(q, d)$.

Several factors should be considered in the Step 1. Different univariate quadrature algorithms have different features. For example, Clenshaw-Curtis algorithm uses ending points, while the Gauss-Patterson algorithm doesn't. If we need to apply inverse normal transformation g, then Gauss-Patterson algorithm is apparently a better choice. On the other hand, Clenshaw-Curtis algorithm is very fast and suitable for situations where online computing of weights and scenarios is needed. We refer readers to Davis and Rabinowita [9], Neumaier [25] and Bungartz and Griebel [4] for discussion on this issue. We also need several iterations of the Step 1-3 with increasing q until convergence is observed.

In the Sparse Grid scenario generation algorithm, as shown in formula (8), the weights are linear combinations of the weights for univariate quadrature rule, which are optimized for polynomial functions. The weights are not equal in general, and could be negative as well. Principally, for the purpose of minimizing the linear approximation error, we should not restrict ourselves to positive weights. This principle is demonstrated by the following optimization problem for a functions space $\mathbb{F}$:

$$\underset{w_k, \xi_k, k=1\dots,K}{minimize} \quad \sup_{\substack{f \in \mathbb{F} \\ \|f\|_\infty \le 1}} \left| \int_{[0,1]^d} f(\xi) d_\xi - \sum_{k=1}^{K} w_k f(\xi_k) \right| \tag{17}$$

$$\xi_k \in [0, 1]^d, k = 1, \dots, K \tag{18}$$

$$w_k \ge 0, k = 1, \dots, K \tag{19}$$

Clearly if we remove the constraint (19), the objective value, hence the approximation error, would decrease or at least stay the same. Note that the polynomial exactness of a univariate Gauss quadrature rule means that it solves the optimization problem (17)- (18) for the corresponding polynomial function space. For multivariate case, the scenarios and knots generated by the Smolyak's multivariate quadrature solves the optimization problem (17)- (18) for the multivariate polynomial functions space specified in Corollary 3.1. For the general function space $\mathcal{W}_d^r$, it is an open question how to solve the optimization problem. In this case, the scenarios and weights generated by the Smolyak's algorithm should be viewed as a suboptimal solution.

We point out that there is a trade-off between optimally choosing negative weights to minimize the linear approximation error in (19), and preserving properties that is conducive to solving the approximated instance. When f is a convex function, a linear approximation with only positive weights leads to a convex function as well. The presence of negative weights in (8) might result in a non-convex approximation function. However, in this paper, we find that the general non-convex solver Knitro 5.1 [40] solved the examples presented in the Section 5. The computational results in 5 show that the sparse grid method achieved high accuracy with few scenarios. Hence, it should motivate future research on developing solution methods for solving optimization problems arising from such approximations.

The scenarios generated by the Sparse Grid algorithm are combinations of abscissa which minimize the linear approximation of univariate polynomial integrand functions. The abscissa are adapted to the order of polynomials, hence in the error estimate of the method in (13), the order of differentiability plays a critical role. The upper bound of the approximation error by the Sparse Grid scenario generation method decreases exponentially with the order of differentiability. In contrast, the scenarios generated by a Quasi Monte Carlo method minimize a certain discrepancy on the unit cube $[0, 1]^d$, see Sobol [38] and Niederreiter [27, 26]. The Quasi Monte Carlo method applies to all integrable function, however it doesn't take into account the differentiability of a smooth integrand function. The upper bound of the approximation error of the Quasi Monte Carlo method has no component of differentiability. This bound as shown in Gerstner and Griebel [16] is:

$$c_d K^{-1} (\log K)^d \tag{20}$$

for some constant c_d.

Table 1 shows error bounds for random vector with $d = 3$. It is clear that for both methods the constants $c_{r,d}$ and c_d play an important rule when K is small, but $\frac{(\log K)^4}{K}$ and $\frac{(\log K)^{2r+2}}{K^r}$ dominate when K is large. To compare the two methods, we have to look at their performance for both small K and large K since we care their practical performance and asymptotic performance as well. When K is small, we compare the dominating constants c_d and $c_{r,d}$. However, explicit expressions for both constants are hard to obtain. Computational study of three examples in Section 5 suggests that for these examples the constant $c_{r,d}$ of the Sparse Grid method is much smaller that the constant c_d of the Sobol sequence. When K is large, if the mixed bounded derivative $r \geq 2$, it seems that the sparse grid method shall outperform Quasi-Monte Carlo method substantially; if $r = 1$, the Quasi Monte Carlo method shall perform slightly better. We observed the described behavior for all three numerical examples in Section 5, which implies that the upper bounds are reasonably tight and describe the actual performance of the two methods.

	$r = 1$	$r \geq 2$
SP	$c_{1,3}\frac{(\log K)^4}{K}$	$c_{2,3}\frac{(\log K)^{2r+2}}{K^r}$
QMC	$c_3\frac{(\log K)^3}{K}$	$c_3\frac{(\log K)^3}{K}$

Table 1: Error Bounds for the Sparse Grid and the Quasi Monte Carlo for $d = 3, r = 1$ and $r \geq 2$.

During the research of solving stochastic programming problem by interior point method [6] and [23], the authors gained high order of differentiability of a general recourse function by regularizing it with barrier functions. This motivated the authors to conduct the current research. In this paper, however, we propose the scenario generation method independently without the concept of regularizing a recourse function with barriers. This is because many problems already have smooth integrand functions without any regularization. In these cases algorithms other than the interior point method proposed by the authors could be used to solve the scenario-based approximation as well.

4 Epi-convergence

In this Section we present the general result regarding the epi-convergence property of the sparse grid scenario generation method for multistage problems. We point out that previous results by King and Wets [21], Donohue [12], and Pennanen [29, 28] do not apply to Sparse Grid method since they all assume positive weights.

Definition 4.1 *(Rockfellar and Wets [35]) $\{f_n\}_{n=1}^{\infty}$ epi-converges to f if for any $x \in \mathbb{R}^n$, there exists a sequence $\{x_n \to x\}$ such that $\overline{\lim_n} f_n(x_n) \leq f(x)$, and for all sequences $\{x_n \to x\}$ the inequality $f(x) \leq \underline{\lim_n} f_n(x_n)$ holds.*

We refer readers to an excellent review by Kall [18], which compares epi-convergence with other classical concepts of convergence. Attouch [1] establishes following important property of an epi-convergent sequence.

Proposition 4.1 *(Attouch [1]) If $\{f_n\}_{n=1}^{\infty}$ epi-converges to a limit f, then $\overline{\lim_n} \inf f_n \leq \inf f$. If $\hat{x}_n \in argmin f_n$, and a subsequence $\{\hat{x}_{n_k}\} \to \hat{x}$, then $f(\hat{x}) = \inf f = lim_{k\to\infty} \inf f_{n_k}$.*

Theorem 4.1 *For finite horizon T-stage problem $\mathcal{P}_T$, if C_0 is closed and bounded, $|\rho_1(x_0, \xi_1)| < \infty$, $\forall \xi_1 \in \Xi_1, x_0 \in C_0$, assume that $\forall x_0 \in C_0$, $\rho_1(x_0, \cdot) : \mathbb{R}^{d_1} \to \mathbb{R} \in \mathcal{W}^r_{d_1}$, $r < \infty$, then*

1. *$Z^* \geq \overline{\lim}_K Z^*_K$,*
2. *if the sequence $\{x_0^K\}_{K=1}^{\infty}$ are solutions of $\{\mathcal{P}_1^K\}_{K=1}^{\infty}$, and if it has a cluster point $\hat{x}_0$, then $\hat{x}_0$ is an optimal solution of $\mathcal{P}_1$ and for a subsequence $\{x_0^{K_q}\}_{q=1}^{\infty}$ converging to $\hat{x}_0$, $\lim_q z^*_{K_q} \to z^*$.*

Proof: Let $f(\xi_0, x_0, \xi_1) = c_0 x_0 \star \rho_1(x_0, \xi_1)$. Note that we assume the $\star$ operator preserves boundness. Since $\rho_1(x_0, \xi_1)$ is bounded and C_0 is closed and bounded, we let $f(\xi_0, x_0, \xi_1) \leq M <$

∞. Let $F(x_0) = \int_{\Xi_1} f(\xi_0, \xi_1, x_0) d_{P(\xi_1)}, \forall x_0 \in C_0$ and $Q_K(x_0) = \sum_{k=1}^K w_k f(\xi_0, \xi_1^k, x_0)$. Since $f(\xi_0, \cdot, x_0) \in \mathcal{W}^r, \forall x_0 \in C_0$, by Theorem 3.1 we have

$$\begin{aligned} |F(x_0) - Q_K(x_0)| &\leq c_{r,d_1} K^{-r} (\log K)^{(d_1-1)(r+1)} \|f(\xi_0, \cdot, x_0)\|_\infty \\ &\leq c_{r,d_1} K^{-r} (\log K)^{(d_1-1)(r+1)} M \to 0, \forall x_0 \in C_0. \end{aligned}$$

Hence, $Q_K(x_0) \to F(x_0)$ uniformly on C_0, i.e., for any convergent sequence $\{x_0^K\}_{k=1}^\infty \to \bar{x}_0 \in C_0$,

$$\forall \epsilon > 0, \exists N_1(\epsilon) : |Q_n(x_0^K) - F(x_0^K)| < \epsilon, \forall n > N_1(\epsilon). \tag{21}$$

Since $F(x_0)$ is continuous on closed and bounded C_0, $F(x_0)$ is uniform continuous on C_0, so

$$\exists d(\epsilon) > 0 : |F(x_0^K) - F(\bar{x}_0)| < \epsilon \text{ if } \|x_0^K - \bar{x}_0\| < d(\epsilon). \tag{22}$$

Since $x_0^K \to \bar{x}_0$, the above inequality holds for all K large enough, say $K \geq N_2(\epsilon)$. So we have

$$|F(\bar{x}_0) - Q_K(x_0^K)| \leq |F(\bar{x}_0) - F(x_0^K)| + |F(x_0^K) - Q_n(x_0^K)| < 2\epsilon, \forall K > \max(N_1(\epsilon), N_2(\epsilon)). \tag{23}$$

Hence, we have continuous convergence of $Q_K(x_0^K) \to F(\bar{x}_0)$. Clearly

$$Q_K(x_0^K) \geq \inf_{x_0 \in C_0} Q_K(\cdot), \tag{24}$$

hence, we have

$$F(\bar{x}_0) = \lim_K Q_K(x_0^K) = \overline{\lim_K} Q_K(x_0^K) \geq \overline{\lim_K} \left(\inf_{x_0 \in C_0} Q_K(\cdot) \right). \tag{25}$$

By taking infimum of the above inequality, we have

$$z^* = \inf_{x_0 \in C_0} F(\cdot) \geq \overline{\lim_K} \left(\inf_{x_0 \in C_0} Q_K(\cdot) \right) = z_K^*. \tag{26}$$

Now let us consider a specific sequence. The solution sequence $\{x_0^K\}_{K=1}^\infty$ of the problem sequence $\{\mathcal{P}_1^K\}_{K=1}^\infty$ might not converge. However if it has a cluster point $\hat{x}_0$, we consider a subsequence $\{x_0^{K_q}\}_{q=1}^\infty \to \hat{x}_0$. By applying (25) to this subsequence, we have

$$F(\hat{x}_0) = \lim_q Q_{K_q}(x_0^{K_q}) = \lim_q \left(\inf_{x_0 \in C_0} Q_{K_q}(\cdot) \right), \tag{27}$$

and since

$$F(\hat{x}_0) \geq \inf_{x_0 \in C_0} F(\cdot), \tag{28}$$

we have

$$\overline{\lim_q} \left(\inf_{x_0 \in C_0} Q_{K_q}(\cdot) \right) \leq \inf_{x_0 \leq C_0} F(\cdot) \leq F(\hat{x}_0) = \underline{\lim_q} \left(\inf_{x_0 \in C_0} Q_{K_q}(\cdot) \right). \tag{29}$$

Hence, $\hat{x}_0$ is an optimal solution of $(\mathcal{P})$, and $\lim_q z_{K_q}^* = z^*$. □

We now comment on Theorem 4.1. For problems with relative completeness assumption as described in Section 2, the boundedness assumption of $|\rho_1(\xi_0, \xi_1, x_0)|$ holds naturally. For a problem instance of interest, we need to know the smoothness of the recourse function $\rho(\xi_0, \cdot, x_0) : \mathbb{R}^{d_1} \rightarrow \mathbb{R} \in \mathcal{W}_{d_1}^r$, $r < \infty$, $\forall x_0 \in C_0$, which is problem dependent. This is exactly the advantage of sparse grid scenario generation algorithm. It improves bounds using the smoothness of the integrand function even when we don't know the smoothness in advance. Below we state results for the static problems and the two-stage linear problems.

Corollary 4.1 *If ($\mathcal{P}$) is a static problem, i.e., $K = 1$, $\xi = [\xi_0, \xi_1]$, $f(\xi, x(\xi)) = f_0(\xi_0, \xi_1, x_0)$, Theorem 4.1 holds if $f_0(\xi_0, \xi_1, x_0) \in \mathcal{W}_d^r$.*

Two-stage linear problems with fixed recourse are well know to have a piecewise linear recourse function, which has weak derivative $r = 1$ by definition, see Walter Rudin [36]. Hence, the requirement of smoothness of the recourse function holds in general for two-stage linear problems.

Corollary 4.2 *If ($\mathcal{P}$) is a two-stage linear problem, i.e., $K = 1$, $\xi = [\xi_0, \xi_1]$, $f(\xi, x(\xi)) = f_0(\xi_0, x_0) + \rho_1(\xi_0, x_0)$, and objectives and constraints are linear, then Theorem 4.1 holds with $r = 1$.*

5 Numerical Examples

In this section we use three examples to demonstrate the performance of the Sparse Grid method: the Markowitz model, the utility maximization model, and the CVaR optimal portfolio model. The dataset for Markowitz model and the CVaR model is taken from Rockafellar and Uryasev [34]. 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 joint normal distribution. The Utility Maximization model has six instruments which have joint normal distribution with a randomly generated mean vector and covariance matrix. The Utility Maximization model was used by Pennanen [32] to compare Quasi Monte Carlo method with Monte Carlo method for generating scenarios for multistage stochastic problems.

We used ANSI C to implement the Sparse Grid scenario generation method as a AMPL user defined function described in Fourer, Gay and Kernighan [15], so we can generate scenarios within AMPL environment. We solve the approximated problems using Knitro 5.1 [40] within AMPL environment with all default settings. All experiments are programmed using AMPL scripts. All computations are performed on a DELL XPS 1210 with a 1.83GHz Intel Core 2 CPU and 1G memory.

We used the Gauss-Patterson univariate quadrature rule to construct multivariate scenarios by sparse grid scenario generation algorithm. We used the Sobol sequence to construct Quasi Monte Carlo scenarios. We are interested in demonstrating the value of the proposed Sparse Grid method by comparing representative methods from both methodologies. A numerical comparison of variants of both methodologies is an ongoing research project.

5.1 Markowitz Model

We first illustrate the polynomial exactness property of the Sparse Grid method using the classic Markowitz model. Pennanen and Koivu [31] used this example for comparing the Quasi Monte Carlo scenario generation method with Monte Carlo method. For a given dataset, the Markowitz model itself is solvable by an quadratic programming solver. The dataset we used is given in the Appendix 5.4. The true optimal value of the model is 0.003785. On the other hand, its formulation involves a multivariate integral, hence, we can approximate it by scenarios. Consequently we can evaluate the quality of scenarios generated by Quasi Monte Carlo and Sparse Grid scenario methods by comparing the objective values of the approximated scenario-based models with the true optimal objective value.

Example 5.1 *(Markowitz model) Let $x = [x_1, \dots, x_K]$ be the amount invested in the K financial instruments, $x_i \geq 0$ and $\sum_{i=1}^{K} x_i = 1$. Let $\xi = [\xi_1, \dots, \xi_K]$ be the random returns of these instruments, $p(\xi)$ be the density of the joint distribution, $m = [m_1, \dots, m_K]$ be the mean return, and $V \in \mathbb{R}^{K \times K}$ be the covariance matrix. We require that the mean return of portfolio x be at least R. And we would like to minimize the portfolio variance. The problem could be represented as:*

$$\begin{aligned} &\textit{minimize } x^T V x \\ &\textit{subject to } |x|_1 \leq 1, x^T \xi \geq R, x \geq 0 \end{aligned} \tag{30}$$

The model (30) is equivalent to:

$$\begin{aligned} &\text{minimize } \int_{\Xi} (\xi^T x - m^T x)^2 p(\xi) d_\xi \\ &\text{subject to } |x|_1 \leq 1, m^T x \geq R, x \geq 0 \end{aligned}$$

Hence, we can approximate it by a scenario-based model as

$$\begin{aligned} &\text{minimize } \sum_{k=1}^{K} w_k (\xi_k^T x_k - m^T x_k)^2 \\ &\text{subject to } \|x_k\|_1 \leq 1, m^T x_k \geq R, x \geq 0 \end{aligned}$$

The objective values of the approximated models are shown in Table 2 and plotted in Figure 2. Note that the true optimal objective value is 0.003785 and we observed it in the table for sample size exceeding 1023, i.e., the linear approximation with scenarios and weights generated by the Sparse Grid method is *exact*. This example supports the conclusion that for stochastic problem with polynomial integrand functions, the Sparse Grid method can achieve exactness with enough number of scenarios. Note that in this example there are three random variables, Sparse Grid method generates $1, 7, 31, 111, 351, 1023$ and 2815 scenarios, which correspond to the parameter value q in (6) being $1, 2, 3, 4, 5, 6$ and 7 respectively. We also apply Quasi Monte Carlo method with the same number of scenarios. It is shown that the Sparse Grid method reached the 1^{st}, 2^{nd}, 3^{rd} and the 4^{th} significant digit at 7, 111, 351, and 1023 number of scenarios. In contrast, the Sobol sequence needed 111 and 1023 number of scenarios to reach the 1^{st} and 2^{2nd} significant digit. We further took samples of size 3000 from Sobol sequence, and then incremented the size by 1000 up to $500,000$. We observed steady but slow convergence. The 3^{rd} significant digit was achieved with $2,300$ scenarios, and the 4^{th} 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 Sobol sequence which is designed for any integrable functions.

K	SP(obj)	QMC(obj)
1	0.000000	0.000000
7	0.003091	0.001099
31	0.003674	0.002990
111	0.003769	0.003398
351	0.003783	0.003641
1023	0.003785	0.003726
2815	0.003785	0.003760

Table 2: Approximated Objective Value of the Markowitz Model. The true objective value is known to be 0.003785.

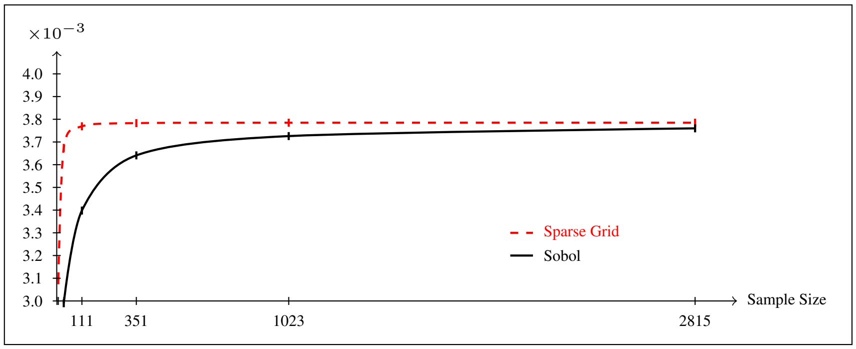


Figure 2: Approximated objective value of the Markowitz Model. The y-axis shows the optimal values from the approximated model for different scenario generation methods for sample size (x-axis) from 1 to 2815. The true optimal value is 3.785×10^{-3}.

5.2 Utility Maximization Model

We are interested in the performance of the Sparse Grid scenario generation method on stochastic problem with non-polynomial integrand functions as well. Hence, we construct the following utility maximization model.

Example 5.2 *(Utility maximization) Let $x = [x_1, \ldots, x_K]$ be the amount invested in the K financial instruments, $x_i \geq 0$ and $\sum_{i=1}^{K} x_i = 1$. Let $\xi = [\xi_1, \ldots, \xi_K]$ be the random returns of these instruments, $p(\xi)$ be the density of the lognormal distribution, $m = [m_1, \ldots, m_K]$ be the mean return. We would like to maximize our exponential utility function. The problem is represented as:*

$$\begin{aligned} &\textit{maximize} \int_{\Xi} -e^{-x^T\xi} p(\xi) d_\xi \\ &\textit{subject to } |x|_1 \leq 1, x \geq 0. \end{aligned}$$

The objective values of the approximated models are shown in Table 3 and Figure 3. There is no analytical solution for the Utility maximization problem. However, both the Sparse Grid method and the Sobol sequence seem to converge to the value -0.0533. The Sparse Grid method seems to converge much faster. It reached the 1^{st}, 2^{nd} and 3^{rd} significant digit with 13, 97, 2561 scenarios. In contrast, the Sobol sequence used 545, and 10625 scenarios to get the 1^{st} and 2^{nd} significant digit. We could not get the 3^{rd} significant digit even after generating $230,000$ scenarios from the Sobol sequence. It is clear that in this example Sparse Grid method was able to use the smoothness of the exponential utility integrand function to achieve faster convergence.

K	SP	QMC
1	-0.028214	-0.028214
13	-0.051023	-0.030882
97	-0.053042	-0.046958
545	-0.053298	-0.052051
2561	-0.053301	-0.052926
10625	-0.053300	-0.053282
40193	-0.053301	-0.053273

Table 3: Approximated objective value of the Utility Maximization Model

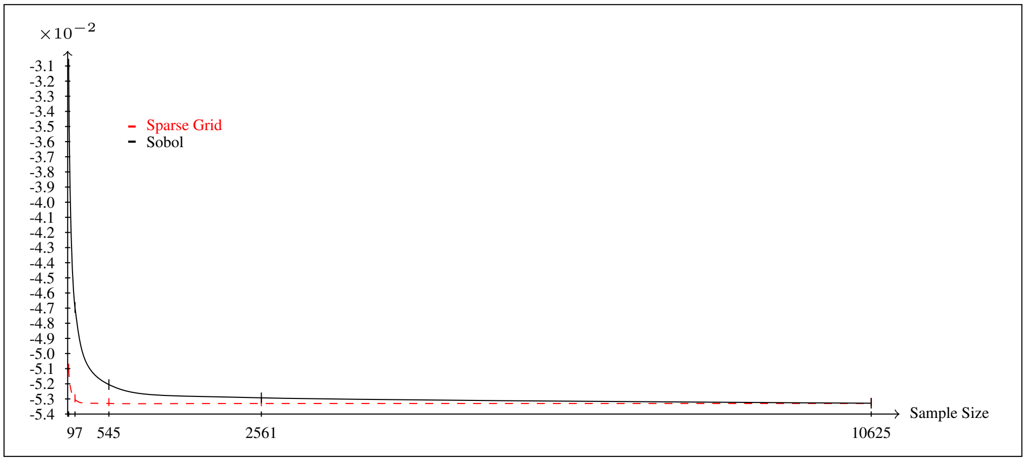


Figure 3: Approximated objective value of the Utility Maximization Model. The y-axis shows the objective values for the approximated mode for different scenario generation methods for various sample size (x-axis).

5.3 CVaR Optimal Portfolio Model

We now study the performance of sparse grid method on the CVaR optimal portfolio model. This problem is modeled as a two-stage linear problem whose recourse function is piecewise linear. The weak derivative of a piecewise linear function is of order one.

Example 5.3 *(CVaR optimized portfolio) Let $x = [x_1, \ldots, x_K]$ be the amount invested in the K financial instruments, $x_i \geq 0$ and $\sum_{i=1}^{K} x_i = 1$. Let $\xi = [\xi_1, \ldots, \xi_K]$ be the random returns of these instruments. $p(\xi)$ be the density of the joint distribution. And $m = [m_1, \ldots, m_K]$ be the mean return. Then the loss function, negative of the yield function is $-x^T\xi$. We require that the mean return of portfolio x be at least R. And we would like to minimize the coherent risk measure β-CVaR for a certain β,* *i.e. 0.9, 0.95, 0.99. By the model of Rockafellar and Uryasev [34], the problem could be represented as:*

$$\begin{aligned}&\textit{minimize } F_\beta(x,\alpha) := \alpha + \frac{1}{1-\beta}\int_{\xi\in\mathbb{R}^n}(-x^T\xi-\alpha)^+p(\xi)d_\xi\\&\textit{subject to } |x_i|_1 \leq 1, x^Tm \geq R, x \geq 0\end{aligned}$$

It is shown by Rockfellar and Uryasev that the optimal objective value is also the β-CVaR of the optimal portfolio x^*, and the β-VaR is the left end point of an interval argmin $F_\beta(x,\alpha)$ which is nonempty, closed and bounded (could be only one point).

Let N be the index set of the scenarios, and let N_1 be the index set of scenarios with positive weights and N_2 be its complement. Let $w_i^+ > 0, w_i^- > 0$. Then the approximated problems is

$$\begin{aligned}&\text{minimize } \tilde{F}_\beta(x,\alpha) := \alpha + \frac{1}{1-\beta}\sum_{i\in N_1} w_i^+(-x^T\xi_i-\alpha)^+ - \frac{1}{1-\beta}\sum_{i\in N_2} w_i^-(-x^T\xi_i-\alpha)^+ \qquad (31)\\&\text{subject to} x_i^Tm \geq R, x_i \geq 0\end{aligned}$$

The approximated problem (31) is equivalent to the following problem after proper dual transformation on subproblems, see Appendix A.

$$\begin{aligned}&\text{minimize } \alpha + \frac{1}{1-\beta}\sum_{i\in N_1} w_i^+ z_i + \frac{1}{1-\beta}\sum_{i\in N_2} w_i^-(u_i x^T\xi_i + u_i\alpha)\\&\text{subject to } |x|_1 \leq 1, x^Tm + \alpha \geq R, i \in N \qquad (32)\\&\qquad z_i + x^T\xi_i + \alpha \geq 0, z_i \geq 0, i \in N_1\\&\qquad 0 \leq u_i \leq 1, i \in N_2\end{aligned}$$

Under normality assumption, the true optimal objective value of the β-CVaR model can be calculated since the optimal solution of the Markowitz model, x, is also optimal for the β-CVaR model when $\beta \geq 0.5$. In this case, both the Var and the CVaR of the optimal portfolio can be computed by the formula:

$$\beta - VaR = -R + c_1(\beta)\sigma(x) \quad \text{and} \quad \beta - CVaR = -R + c_2(\beta)\sigma(x), \qquad (33)$$

where the $c_1(\beta)$ and $c_2(\beta)$ are

$$c_1(\beta) = \sqrt(2)erf^{-1}(2\beta-1) \quad \text{and} \quad c_2(\beta) = \left(\sqrt{2\pi}\exp(erf^{-1}(2\beta-1))^2(1-\beta)\right)^{-1}, \qquad (34)$$

and the $erf^{-1}(z)$ means the inverse of the error function

$$erf(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{1t^2}dt. \qquad (35)$$

This provides us an opportunity to compare an approximated optimal objective value with the true optimal objective value. We use the same dataset of the Markowitz model Example (30). Let z be the

CVaR value of the approximated model (32), and z^* be the true CVaR value, we define the percentage difference of the CVaR value as:

$$\text{pdiff} = \frac{z - z^*}{z^*}. \tag{36}$$

The percentage difference values for the Sparse Grid method and Sobol sequence are shown in Table 4 and plotted in the Figure 4. In this case, the limit of the approximation is 0. We observed that for all βs, Sparse Grid method reached the $1st$ and 2^{nd} significant digit much more rapidly than Sobol sequence. However, the two methods seemed to require similar number of scenarios to gain more precise result. In most applications, it is important to get a reasonably precise approximation early, i.e., 1% percentage difference in this example.

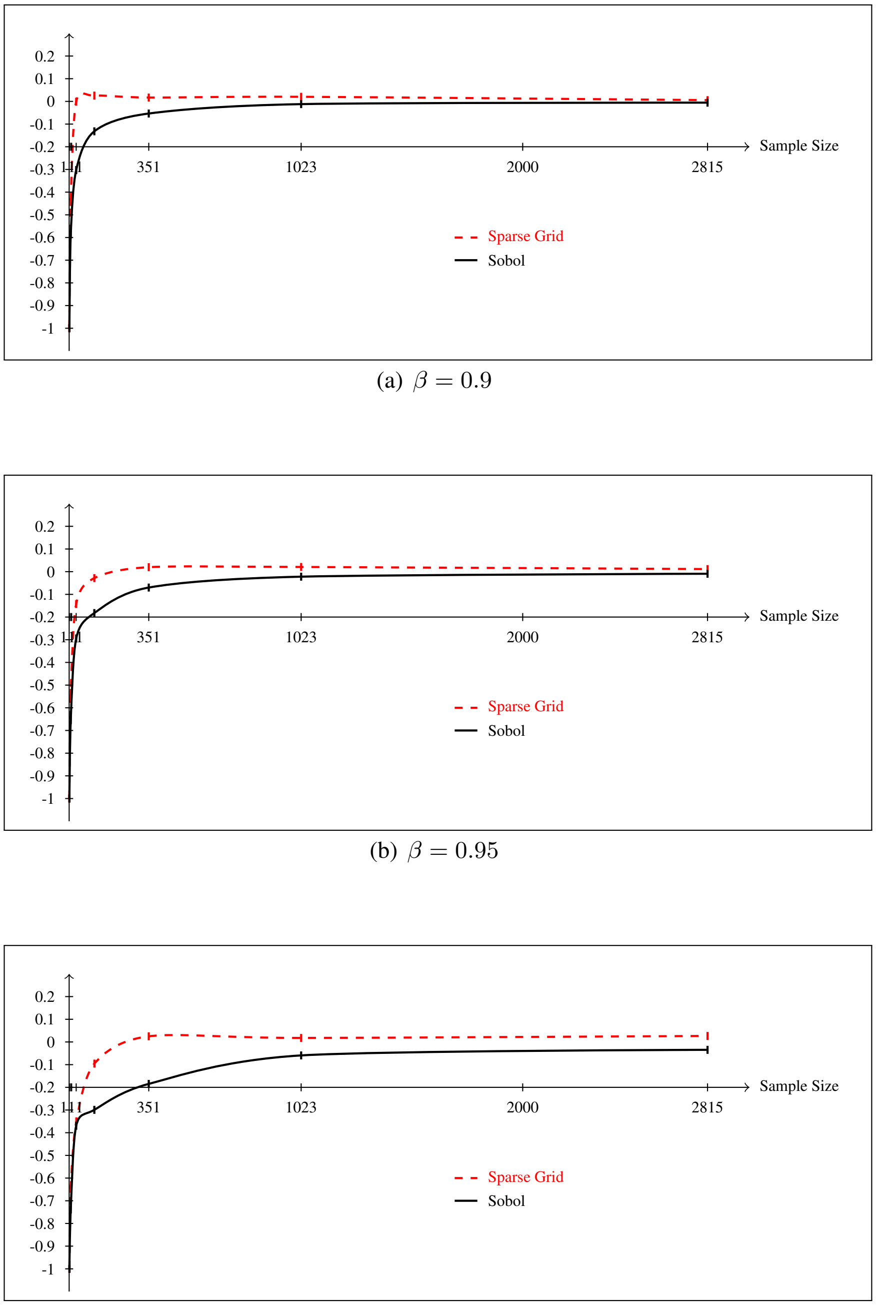
	pdiff($\beta = 0.9$)		pdiff($\beta = 0.95$)		pdiff($\beta = 0.99$)	
K	SP	QMC	SP	QMC	SP	QMC
1	-0.999998	-1	-0.999997	-1	-0.99998	-1
7	-0.443303	-0.587115	-0.534228	-0.654558	-0.64704	-0.738264
31	-0.00120	-0.302735	-0.142668	-0.294612	-0.35023	-0.369318
111	0.025581	-0.131899	-0.028402	-0.18241	-0.09539	-0.299214
351	0.016986	-0.053489	0.019736	-0.069991	0.024924	-0.184810
1023	0.020240	-0.012058	0.020723	-0.022002	0.017752	-0.059259
2815	0.005442	-0.005566	0.011380	-0.009114	0.026499	-0.034649

Table 4: Percentage difference in objective value of the CVaR Optimal Portfolio Model for $\beta = 0.9, 0.95, 0.99$.

5.4 Discussion of Numerical Examples

By the three test examples, we observe that for problems with smooth integrand functions, i.e., $r \geq 2$, the Sparse Grid method converges quickly, and also has a superior superior empirical convergence rate at tail. For the problem with less smooth integrand function, i.e., $r = 1$, the Sparse Grid method had similar performance at tail comparing with the Sobol sequence, but still converges to the right value very quickly. The error estimates for the Sparse Grid in (14) and Quasi Monte Carlo (20) may provide an explanation. Table 1 suggests that if in the CVaR portfolio model, $c_3 > c_{1,3}$, then Spare Grid method shall better performance when K is small. However when K is large, the Quasi Monte Carlo method is likely to perform better for this case with $r = 1$. Table 1 also clearly explains that for $r \geq 2$, the Sparse Grid method is superior due to the dominating exponential factor $\frac{1}{K^r}$.

Acknowledgment We thank Professor John Birge and Tito Homme-de-Mello for useful discussion and suggestions. We appreciate helps from Professor Robert Fourier and David Gay on implementing AMPL user defined functions of the Sparse Grid scenario generation method. We thank Professor Jorge Nocedal for providing us the Knitro 5.1, which we used to solve all numerical examples.



(a) $\beta = 0.9$

(b) $\beta = 0.95$

(c) $\beta = 0.99$

Figure 4: The percentage difference in objective value of CVaR Optimal Portfolio Model for $\beta = 0.9, 0.95, 0.99$.

Appendix A: Extensive Formulation of a Two-stage Linear Problem with Negative Scenario Weights

We show a useful transformation within a scenario-based two-stage linear problem to convert it into its extensive formulation when there are both positive and negative scenario weights. We model a two-stage linear problem with fixed recourse as the following.

$$\begin{aligned}&\text{minimize } c_0^T x_0 + \int_{\Xi_1} \rho(\xi_0, \xi_1, x_0) d_{P(\xi_1)}\\ &\text{subject to } x_0 \in C_0 = \{x_0 | A_0 x_0 = b_0, x_0 \geq 0\}\end{aligned} \qquad (\mathcal{P}_2)$$

where $\rho(\xi_0, \xi_1, x_0)$ is defined to be

$$\begin{aligned}&\text{minimize } c_1^T x_1\\ &\text{subject to } x_0 \in C_1 = \{x_1 | A_1 x_1 = b_1 - T_1 x_0, x_1 \geq 0\}\\ &\qquad \xi = [c_1, b_1, T_1]\end{aligned}$$

Under assumptions in Section 2, the problem is bounded and feasible, and achieves optimal objective value z^*.

Let $\Xi_1 \in \mathbb{R}^{d_1}$ and $\{\xi_1^k\}_{k=1}^K$ be a set of generated scenarios, then the corresponding approximated problem is:

$$\begin{aligned}&\text{minimize } c_0^T x_0 + \sum_{k=1}^{K} w_k \rho(\xi_0, \xi_1^k, x_0) d_{P(\xi_1)}\\ &\text{subject to } x_0 \in C_0 = \{x_0 | A_0 x_0 = b_0, x_0 \geq 0\},\end{aligned} \qquad (\mathcal{P}_2^K)$$

where $\rho(\xi_0, \xi_1^k, x_0)$ is defined as

$$\begin{aligned}&\text{minimize } {c_1^k}^T x_1^k\\ &\text{subject to } x_1^k \in C_1^k = \left\{x_1 | A_1 x_1^k = b_1^k - T_1^k x_0, x_1^k \geq 0\right\}\end{aligned}$$

Under assumptions in Section 2, the approximated problem is also bounded and feasible, and achieves optimal objective value z_K^*.

Let K_1 and K_2 be index sets of positive weights w_k^+ and negative weights $-w_k^-$. We reformulate $(\mathcal{P}_2^K)$ as:

$$\begin{aligned}&\text{minimize } c_0^T x_0 + \sum_{k \in K_1} w_k^+ c_1^k x_1^k - \sum_{k \in K_2} w_k^- {u_1^k}^T (b_1^k - T_1^k x_0)\\ &\text{subject to } A_0 x_0 = b_0, x_0 \geq 0\\ &\qquad A_1 x_1^k = b_1^k - T_1^k x_0, x_1^k \geq 0, k \in K_1\\ &\qquad {u_1^k}^T A_1 \leq c_1^k, k \in K_2.\end{aligned} \qquad (\mathcal{PD}_2^K)$$

Note that in order to achieve optimal approximation of the integral, the sparse grid method searches optimal weights in $\mathbb{R}$, instead of its subset $\mathbb{R}^+$.

Theorem 5.1 *If $\mathcal{P}_2^K$ is feasible and bounded, then $\mathcal{PD}_2^K$ is feasible and bounded and has the same objective value.*

Proof: We form dual of $\rho(\xi_0, \xi_1^k, x_0)$ as $\phi(\xi_0, \xi_1^k, x_0)$:

$$\begin{aligned}&\text{maximize } {u_1^k}^T(-b_1^k + T_1^k x_0)\\ &\text{subject to } u_1^k \in U_1^k = \left\{u_1^k | {u_1^k}^T A_1 \leq c_1^k\right\}\end{aligned}$$

Under our assumption of relative complete recourse and by duality theory of linear programming we have $\rho(\xi_0, \xi_1^k, x_0) = \phi(\xi_0, \xi_1^k, x_0)$. So the objective function of $(\mathcal{P}_2^K)$ could be written equivalently as

$$J(x_0) \overset{def}{=} c_0^T x_0 + \sum_{k \in K_1} w_k^+ \rho(\xi_0, \xi_1^k, x_0) d_{P(\xi_1)} - \sum_{k \in K_2} w_k^+ \phi(\xi_0, \xi_1^k, x_0) d_{P(\xi_1)}, \tag{A-1}$$

and we let the objective function of $\mathcal{PD}_2^K$ be $H(x_0, x_{K_1}, u_{K_2})$, where x_{K_1} means vector of $x_1^k, \forall k \in K_1$ and u_{K_2} means vector of $u_1^k, \forall k \in K_2$.

Let $x_0 \in C_0$, $x_1^k \in C_1^k, \forall k \in K_1$ and $u_1^k \in U_1^k, \forall k \in K_2$. Since $c_1^k x_1^k \geq \rho(\xi_0, \xi_1^k, x_0)$ and ${u_1^k}^T(-b_1^k + T_1^k x_0) \leq \phi(\xi_0, \xi_1^k, x_0)$, we have

$$w_k^+ c_1^k x_1^k \geq w_k^+ \rho(\xi_0, \xi_1^k, x_0), \ -w_k^- {u_1^k}^T(-b_1^k + T_1^k x_0) \geq -w_k^- \phi(\xi_0, \xi_1^k, x_0) \tag{A-2}$$

and hence,

$$H(x_0, x_{K_1}, u_{K_2}) \geq J(x_0) \geq \text{obj}_{\mathcal{P}_2^K}. \tag{A-3}$$

Hence, the minimal $H(x_0, x_{K_1}, u_{K_2})$ over its feasible region is also greater than or equal to $\text{obj}_{\mathcal{P}_2^K}$. The equality can be achieved by setting (x_0, x_{K_1}, u_{K_2}) to be the optimal solution of $\mathcal{P}_2^K$. □

Appendix B: Dataset

Table 5 and 6 are used in the Markowitz model and the CVaR optimal portfolio model. In both models, the minimum return R is 0.011. Note that the dataset is same as the one used by Rockafella and Uryasev [34]. The random generated data in Table 7 and 8 are used in the Utility Maximization model.

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

Table 5: 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 6: Portfolio Covariance Matrix

Stock	Mean Return
1	1.186573284
2	1.271987547
3	1.262317169
4	1.165045884
5	1.050464674
6	0.990708953

Table 7: Stock Mean Return

	1	2	3	4	5	6
1	0.171633	0.269857	0.108958	0.172031	0.127180	0.234007
2	0.158851	0.161127	0.157614	0.189621	0.143484	0.295287
3	0.237658	0.245012	0.255039	0.261027	0.227806	0.127998
4	0.105797	0.186469	0.254520	0.165738	0.213629	0.193843
5	0.278903	0.142303	0.275562	0.217241	0.175747	0.110844
6	0.242071	0.165333	0.217824	0.234243	0.104976	0.276837

Table 8: Square Root of Stock Covariance Matrix

References

[1] H. Attouch. *Variational Convergence For Functions And Operators*. Pitman (Advanced Publishing Programs), 1984.

[2] Volker Barthelmann, Erich Novak, and Klaus Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mathematics*, V12(4):273–288, March 2000.

[3] John R. Birge and Francois Louveaux. *Introduction To Stochastic Programming*. Springer, New York, 1997.

[4] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta numerica*, pages 147–269, 2004.

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

[6] Michael Chen and Sanjay Mehrotra. Self-concordant tree and decomposition based interior point methods for stochastic convex optimization problem. *Stochastic Programming E-Print Series*, 2007-8, 2007.

[7] A. Chiralaksanakul and D. Morton. Assessing policy quality in multistage stochastic programming. *Stochastic Programming E-Print Series*, 12, 2004.

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

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

[10] 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.

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

[12] 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.

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

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

[15] Robert Fourer, David M. Gay, and Brian W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press / Brooks/Cole Publishing Company, 2002.

[16] T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, V71(1):65–87, August 2003.

[17] Holger Heitsch and Werner Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 2007.

[18] Peter Kall. Approximation to optimization problems: An elementary review. *Mathematics Of Operations Research*, 11:9–18, 1998.

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

[20] Michal Kaut and Stein W. Wallace. Shape-based scenario generation using copulas. *Working Paper*, 2006.

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

[22] Averill M. Law and W. David Kelton. *Simulation modeling and anaylsis*. Tata McGraw-Hill, 2000.

[23] Sanjay Mehrotra and M. Gokhan Ozevin. Convergence analysis of a weighted barrier decomposition algorithm for two stage stochastic programming. Technical Report 07-07, IEMS, Northwestern University, 2007.

[24] Kjetil Hyland Michal Kaut and Stein W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24:169–185, 2003.

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

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

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

[28] T. Pennanen. Epi-convergent discretizations of multistage stochastic programs. *Mathematics of Operations Research*, 30(1):245–256, 2005.

[29] T. Pennanen. Epi-convergent discretizations of multistage stochastic programs via integration quadratures. *Mathematical Programming Series B, to appear*, 2007.

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

[31] T. Pennanen and M. Koivu. Epi-convergent discretization of stochastic programs via integration quadratures. *Numerische Mathematik*, 100(1):141–163, 2005.

[32] Teemu Pennanen and Petri Hilli. Numerical study of discretization of multistage stochastic programs. http://math.tkk.fi/ teemu/publications.html, 2006.

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

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

[35] R.T. Rockafellar and R.J.-B Wets. *Variational Analysis*. Springer-Verlag, 1998.

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

[37] 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)).

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

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

[40] Richard A. Waltz and Todd D. Plantenga. *Knitro 5.1 User's Manual*. Ziena Optimization, Inc, www.ziena.com, 2007.

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