An Ordinal-Optimization-based Approach to Stochastic Mixed Integer Nonlinear Programming

Chengtao Wen, B. Erik Ydstie, Pittsburgh, PA, US

Introduction

The stochastic Mixed Integer Nonlinear Programming (MINLP) is an important and fundamental problem in the area of process system engineering. This has motivated the development of efficient algorithms for handling the large-scale, non-convex, highly combinatorial and strongly nonlinear problems.

The methods to solve the stochastic MINLP problems fall into two categories: numerical integration method and sample average method. The numerical integration method utilizes numerical integration techniques to calculate the multi-dimensional integrals, such as the Gaussian Quadratures/Cubatures method. The Monte Carlo simulation is one of the most widely used sample average methods. Acevedo and Pistikopoulos (1998) suggested the use of numerical integration methods for smaller dimension problems and sampling-based methods for larger problems [1].

The established algorithms to stochastic MINLP problems are successful in many applications. However, they are built on cardinal optimization, in which the optimums are obtained from the accurate calculation of cardinal values. The cardinal optimization is generally associated with inherent computational difficulties from the combinatorial natural of discrete variables and stochastic natural of uncertain parameters. It is easy to become intractable with the increase of the number of discrete variables and/or uncertain parameters.

The ordinal optimization (OO) is a novel optimization methodology [2]. It is designed to cope with hard problems such as problems with uncertainties, or problems with huge sample space that grows exponentially with respect to the problem size. The OO theory provides a way to obtain solutions of high quality with much less computation effort than the conventional cardinal optimization methods.

The OO theory is based on the following two tenets: 1) it is much easier to determine “order" than “value.” To determine whether *A* is better or worse than B is a simpler task than to determine how much better is *A* than *B* (i.e., the value of *A*-*B*) especially when uncertainties exist. 2) Instead of asking the “best for sure,” we seek the “good enough” solution with high probability. This goal softening technique makes the optimization problem much easier. The OO method has found successful applications in a variety of fields, e.g. optimal control, network topology design, planning and scheduling of manufacturing systems.

In this paper, an ordinal-optimization-based algorithm is developed to solve the stochastic MINLP problems. A novel approximation method is proposed, in which the expected performances are estimated by the intermediate NLP optimization solutions and Monte-Carlo simulations with a small sample size. The OO theory is then applied to form a subset of good enough decisions with high probability by ranking the crude performance estimations. The sub-optimal solution is obtained by

searching the subset of selected good decisions and using the accurate performance values. A benchmark problem is illustrated to show the extra high computation efficiency and the quality of the obtained solutions.

Stochastic MINLP Problems

In this paper, we consider the following stochastic MINLP problem

$$\min_{x,y} E_\theta[f(x,y,\theta)] \tag{1}$$

$$s.t. \quad \begin{cases} h(x,y,\theta) = 0 \\ g(x,y,\theta) \le 0 \end{cases} \tag{2}$$

where $x \in \Re^m$ is the continuous variables, $y \in Z^n$ represents the discrete variables, and $\theta \in \Re^k$ is the uncertain parameters. The *h* and *g* denote a set of equality and inequality constraints, respectively. The scalar function *f* represents the objective function. Typically, it is an expected value of an economic performance. For simplicity, we define *y* as the decision variable and *x* the operation variable. The objective of a stochastic MINLP problem is then described as finding the optimal decision and operation variables in presence of uncertain conditions.

Ordinal Optimization

The OO theory is based on two central ideas: order comparison and goal softening.

The order comparison is to evaluate the quality of the expected performances with their relative order rather than the exact cardinal values. According to [3], the relative order of expected performances converges exponentially fast. By comparison, the cardinal values converge at rate of $\frac{1}{\sqrt{N}}$. The order converges much faster than the cardinal values with respect to the number of samples.

The second basis is goal softening. The OO settles for the "good" solutions with high probability. This deviates from the conventional optimization algorithms, which ask for the best solution for sure. By softening our goal to the "good" solutions, e.g. top-*g* solutions in the search space, the number of good solutions in the set of estimated top-g decisions can be quite substantial even in presence of very large estimation errors in the performance values.

Suppose we simultaneously evaluate a large set of alternatives very approximately and order them according to the approximate evaluations. High probability can be anticipated that we can find the actual good alternatives by limiting ourselves to the top of the observed good choices.

Ordinal-Optimization-Based Algorithm to Stochastic MINLP

Approximation of Expected Performances

In order to relieve the computation burden, two methods are used to approximate the expected performance. The first method is a direct heritage from the classical OO theory. The short Monte Carlo simulation is used, whose sample size is at the magnitude of 10^2, i.e. $O(10^2)$. This is insufficient from

the cardinal optimization point of view. But it is often enough for rank comparison.

The second method is to use the NLP solvers with a fixed number of iterations. This is a novel goal softening strategy. Instead of running a NLP solver until it converges to the optimum, we ask for an intermediate optimization results that are "good" enough with high probability. The intermediate results are defined as "good" if it contains enough information for rank comparison.

Theorem 1: Assume that the objective function $f(x \mid y_i)$ is Lipschitz continuous for a fixed y_i with $i \in \{1, \cdots, m\}$. Denote M_i^* as the number of iterations needed to calculate the optimum x^* of $f(x \mid y_i)$ with. Then there must exist *m* positive integers $M_i \le M_i^*$, such that

$$\min_{1 \le i \le m} \left\{ f\left(x^{M_i} \mid y_i\right) \right\} = \min_{1 \le i \le m} \left\{ f\left(x^* \mid y_i\right) \right\} \tag{3}$$

where x^{M_i} is the intermediate optimal results of $f(x \mid y_i)$ obtained from M_i iterations.

Proof: Please refer to [4] for detail proof.

Theorem 1 shows that the optimal decision can be obtained with less computation cost if the intermediate optimization results are fully utilized in term of the rank-based comparison. Using the order comparison and goal softening, the number of iterations of the NLP problems can be much decreased without affecting the final optimum significantly, because the order is very robust to approximation errors [3]. Therefore, the use of intermediate optimizations can lead to a significant reduction in computation burden when the time saving of a single NLP optimization is amplified by the sample size.

Selection of Good Enough Decisions

Given a representative set Π with *M* decisions, whose performances are estimated by using the approximation algorithm stated in previous section. Denote the true and crude expected performances as J_i and $\hat{J}_i$, respectively. We can get

$$J_i = \hat{J}_i + e_i \tag{4}$$

where e_i is the approximation error with $i = 1, \cdots, M$. There may be significant errors due to the rough approximations. The advantage of an OO-based method is its capability to separate the good from the bad even using very crude approximation. The performance "order" is relatively immune to large approximation errors. Therefore, even if the rough estimation is used to rank *M* selections' performances, some good enough decisions will be kept within the select set with high probability.

The major task in applying OO theory is to construct a selected subset *S* containing "good enough" decisions with high probability. The quantitative measure of the "good enough" is the alignment probability defined as

$$P(G, S, a) = P_A\left(\mid G \cap S \mid \ge a\right) \tag{5}$$

where *G* is the "good enough" set and a is called the alignment level. Intuitively the alignment probability is the probability of the event that there are at least *a* elements in the intersection set of *G* and *S*.

To select *s* good ones from *M* representatives, the crude approximated performances are calculated

and ranked. The top *s* decisions are then selected as *S* and its size *s* is determined by a regressed nonlinear equation to satisfy certain confidence requirement. The value of *s* can be estimated by

$$s \approx Z(a,g) = e^{Z_0} a^{\rho} g^{\gamma} + \eta \tag{6}$$

where $g = |G|$ is the size of *G*, and Z_0, ρ, γ, η are coefficients or parameters obtained by nonlinear regression.

The evaluation of *M* decisions using crude approximations is computationally efficient and the OO theory guarantees that good decisions will be among the selected set in high probability. More accurate but time-consuming evaluation is applied to evaluate the selected decisions. For each decision, the accurate performance is calculated using a long Monte-Carlo simulation and the real optimums of NLP problems. The best decision is then selected by evaluating those decisions in the selected subset based on the accurate performances. Since $s = |S|$ is much smaller than *M*, the OO method is extremely efficient in comparison with brute and force method by calculating *M* accurate performances.

Rank-Filter-Based Algorithm

The main steps of the rank-filter-based algorithm are summarized as follows:

1. Choose *M* decisions from the decision space randomly;
2. Run short simulations with a small sample size of N_1 for each decision;
3. Run the NLP solver with *k* iterations for each sample realization;
4. Calculate the approximated expectations of *M* decisions using the average of N_1 intermediate optimization solutions;
5. Rank the approximated expectations and choose the top-*s* ones, where *s* is calculated from (26);
6. Calculate the accurate performances of the selected s decisions using a big sample size of N_2 and optimal NLP solutions;
7. Get the sub-optimal solution by ranking the accurate performances of s decisions.

The parameters in this algorithm can be obtained from the corresponding deterministic MINLP problem, where the random parameters are substituted by their mathematical expectations. A typical set of parameters are shown in [4].

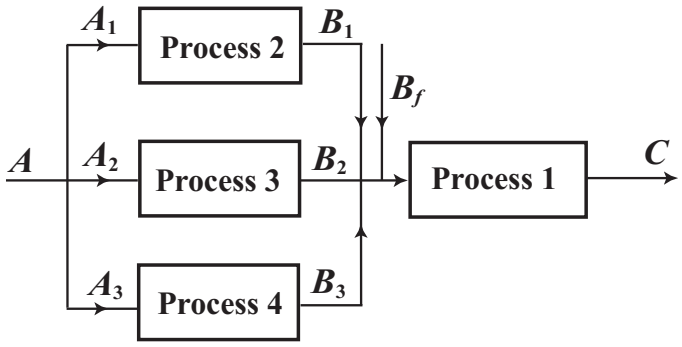

Fig. 1 Process superstructure

Numerical Examples

The example is a process flow-sheet synthesis problem, whose superstructure is shown in Fig. 1.

This problem is also studied by [5] and [6]. In this example, process 1 is fixed. This implies that 3 binary variables are needed to represent the existence of the process 2, 3 and 4. This example involves 7 random parameters. The normal distributions are assumed for 4 continuous random variables, i.e. the availability of raw material *A*, the demand of product *C*, the prices of material *A* and pure *B*(B_f). Refer to [4] for the distribution data of 3 discrete random parameters, i.e. the kinetic constants of process 2, 3 and 4.

$$\max_{x,y} E_\theta\{550x_{10} - (\theta_3 x_1 + \theta_4 x_9) - [100 + 15(x_5 + x_9) + (80 + 5x_2) + (130y_3 + 15x_3) + (150y_4 + 5x_4)]\}$$

$$\begin{aligned}
s.t. \quad & x_1 = x_2 + x_3 + x_4 \\
& x_5 = x_6 + x_7 + x_8 \\
& x_{10} = 0.9(x_5 + x_9) \\
& x_6 = 18\ln(1 + \frac{x_2}{\theta_5}) \\
& x_7 = 20\ln(1 + \frac{x_3}{\theta_6}) \\
& x_8 = 15\ln(1 + \frac{x_4}{\theta_7}) \\
& x_1 \le \theta_2, \quad 5 \le x_2 \le 25y_2 \\
& x_3 \le 20y_3, x_4 \le 20y_4 \\
& x_9 \le 15, \quad x_{10} \le \theta_1 \\
& x \in \Re^{10}, \quad y \in \{0,1\}^3 \\
& \theta_1 \in N(17.5, 2.5) \\
& \theta_2 \in N(27.5, 2.5) \\
& \theta_3 \in N(250, 10) \\
& \theta_4 \in N(300, 15)
\end{aligned}$$

with $x = [A, A_2, A_3, A_4, B, B_2, B_3, B_4, B_f, C]^T$ and $y = [y_2, y_3, y_4]^T$.

The realizations of continuous random parameters are generated using a truncated range of $\mu \pm 3\delta$. To calculate the coarse expected performances, a sample size of 200 is used. The intermediate optimization results are calculated by running a Matlab NLP solver with 15 iterations. The top-2 designs are chosen to form the selected set *S*. The precise expected performances of the selected designs are calculated from 5000 optimums of NLP problems.

Table 1 lists the calculation results of the top-2 decisions. By comparing the accurate optimums, we can get the optimal decision vector $y^* = [1,1,1]^T$, with an expected performance of 4068.6. The same optimal decision is obtained by the decomposition-based algorithm in [6]. The overall calculation time is 282.7 CPUs. It is of high computation efficiency by considering the big sample size of 5000 for calculating the accurate expectations.

Table 1. Expected Performance of Selected Decisions

#	*y*	**Approximated Optimal**	**Real Optimal**
1	[1 1 1]	3848.3	4068.6
2	[1 1 0]	3873.2	4057.6

Conclusion

In this paper, we propose an ordinal-optimization-based algorithm to solve the stochastic MINLP problems. This algorithm is based on a novel approximation method and the OO-theory framework. It guarantees a sub-optimal solution of high quality with high probability. The proposed algorithm can ease several order of magnitude of computation expense without significant loss of solution precision compared to the conventional algorithms based on cardinal optimization. The ordinal-optimization-based algorithm is promising to find successful applications in a variety fields, such as the synthesis of large-scale chemical process, optimization of supply chain, and control of hybrid systems.

Reference

[1] Acevedo, J., & Pistikopoulos, E. N., "Stochastic optimization based algorithms for process synthesis under uncertainty." *Computers and Chemical Engineering*, vol. 22: 647-671, 1998.

[2] Ho, Y. C. "An explanation of ordinal optimization: Soft computing for hard problems." *Information Sciences*, 113(3-4), 169-192, 1999.

[3] Dai, L. Y., "Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems." *Journal of Optimization Theory and Applications*, 91 (2): 363-388, 1996.

[4] Wen, Chengtao, & Ydstie, B. Erik, " An ordinal-optimization-based approach to stochastic MINLP programming." *AIChE Journal* (second submission).

[5] Duran, M. A., & Grossmann, I. E., "An outer-approximation algorithm for a class of mixed integer nonlinear programs." *Mathematical Programming*, 36, 307, 1986.

[6] Goyal, V. & Ierapetritou, M. G., "Stochastic MINLP optimization using simplicial approximation." *Computers and Chemical Engineering*, vol. 31: 1081-1087, 2007.