25

Discretization of a Random Field – a Multiobjective Algorithm Approach

Guang-Yih Sheu
Chang-Jung Christian University
Taiwan

1. Introduction

Discretization of a random field by the generalized polynomial chaos (GPC) begins with selecting a specific type of orthogonal polynomial (e.g. Legendre and Hermite polynomials) (e.g. Ghanem and Spanos, 1991). This selection of a type of orthogonal polynomial can be performed based on the reported experiences (e.g. Xiu and Karniadakis, 2003) or data revealing the distribution of a random field to be discretized. If such data or reported experiences are unavailable, a third way may be generating some pilot tests to study the performance of a specific type of orthogonal polynomial in discretizing this random field. This study tries to develop an evolutionary algorithm-based auxiliary tool for the implementation of such pilot tests. A similar tool (Allaix and Carbone, 2009), which is based on the single-objective evolutionary algorithm, had been developed for constructing the Karhunen-Loève (KL) representation of a random field. Both KL and GPC expansions are two of the popular random field discretization methods (Ghanem and Spanos, 1991). But, the KL expansion should be applied under a prerequisite of knowing the covariance matrix of a random field to be discretized (e.g. Ghanem and Spanos, 1991); while, the GPC expansions can be applied without similar prerequisites (e.g. Xiu and Karniadakis, 2003). Therefore, the development of an auxiliary tool for constructing a GPC representation of a random field would be necessary.

The succeeding research considers the derivation of an GPC representation of a random field as a multi-objective (MO) problem having two goals: (a) limiting the computational efforts spent in applying the resulting GPC representation; and (b) keeping the resulting GPC representation satisfying all accuracy standards (e.g. getting the sufficiently accurate prediction of statistical parameters of a random field). The former goal will be attained by limiting the highest order of polynomial term and total number of uncorrelated random variables used to construct a GPC representation; while the latter goal will be attained by minimizing multiple error estimators. Since there are multiple goals to be attained, a multiple objective evolutionary algorithm (MOEA) is required. Among all available MOEAs, the strength Pareto evolutionary algorithm II (SPEA 2) (Zitzler, et al., 2001) is chosen. The highest order of polynomial term and total number of uncorrelated random variables used to construct an GPC representation are considered as two parameters to be identified.

The remainder of this study is organized into four sections. In Sec. 2, the theoretical backgrounds of GPC expansions (e.g. Xiu and Karniadakis, 2003) are briefly reviewed. In Sec. 3, the SPEA 2 (Zitzler, et al., 2001) is used to construct a parameter identification procedure to identify the aforementioned highest order of orthogonal polynomial and total

number of uncorrelated random variables. Two examples are introduced to study the performance of the resulting works are tested in Sec. 4. The test results are used to give some discussion and conclusion in Sec. 5.

2. Generalized polynomial chaos expansion

2.1 Definition

The GPC (e.g. Xiu and Karniadakis, 2002) is a generalization of the classical Wiener's PC (polynomial chaos). This Wiener's PC is defined as the span of Hermite polynomials of a Gaussian process. A Cameron-Martin theorem states that the Wiener's PC can be used to approximate any functional in $L_2(C)$ and converges in the $L_2(C)$ sense in which C denotes the space of real functions, which are continuous over the interval [0, 1] and vanish at 0. The GPC further provides a mean of expanding second-order random fields having finite variance over a specific interval in terms of orthogonal polynomials. Most physical processes can be simulated by such second-order random fields. As compared to the Wiener's PC, the GPC has better performance in representing some specific types of non-Gaussian inputs.

Suppose θ is an event in the probabilistic space, and u(θ) is a continuous function of θ. The GPC representation of u is equated by (e.g. Xiu and Karniadakis, 2002)

$$\begin{aligned} u = a_0\psi_0(\xi_{i_0}) + \sum_{i_1=1}^{\infty} a_{i_1}\psi_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2}\psi_2(\xi_{i_1},\xi_{i_2}) \\ + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3}\psi_3(\xi_{i_1},\xi_{i_2},\xi_{i_3}) + ... \end{aligned} \tag{1}$$

where a_0, a_{i_1}, $a_{i_1 i_2}$,.... denote the coefficients to be determined and ψ_n, n = 0, 1, 2… are the polynomial chaos (PC) of order n of multi-dimensional independent random variables $\xi_{i_1}, \xi_{i_2} ... and\ \xi_{i_n}$ having zero mean and unit variance.

For the notational convenience, Eq. (1) is further modified to (e.g. Xiu and Karniadakis, 2002)

$$u = \sum_{i=0}^{\infty} U_i \Psi_i(\boldsymbol{\xi}) \tag{2}$$

where $\boldsymbol{\xi} = (\xi_{i_1}, \xi_{i_2} ... \xi_{i_n})$, $\Psi_0 = 1$, and U_0 is set to the mean value of u. There is a one-to-one correspondence between Ψ_i and ψ_n and between a_0, a_{i_1}, $a_{i_1 i_2}$,.... and U_i. For example, suppose u is as functions of **x** and $\boldsymbol{\xi} = (\xi_{i_1}, \xi_{i_2})$. The relationship between Ψ_i, i = 0-9 and ψ_n, n = 0-3 is given by

$$\begin{aligned} &\Psi_0 = \psi_0(\xi_{i_1})\psi_0(\xi_{i_2}) = 1, \quad \Psi_1 = \psi_1(\xi_{i_1})\psi_0(\xi_{i_2}), \quad \Psi_2 = \psi_0(\xi_{i_1})\psi_1(\xi_{i_2}) \\ &\Psi_3 = \psi_2(\xi_{i_1})\psi_0(\xi_{i_2}), \quad \Psi_4 = \psi_1(\xi_{i_1})\psi_1(\xi_{i_2}), \quad \Psi_5 = \psi_0(\xi_{i_1})\psi_2(\xi_{i_2}) \\ &\Psi_6 = \psi_3(\xi_{i_1})\psi_0(\xi_{i_2}), \quad \Psi_7 = \psi_2(\xi_{i_1})\psi_1(\xi_{i_2}), \quad \Psi_8 = \psi_1(\xi_{i_1})\psi_2(\xi_{i_2}) \\ &\Psi_9 = \psi_0(\xi_{i_1})\psi_3(\xi_{i_2}) \end{aligned} \tag{3}$$

In addition, because Ψ_i, i = 0-∞ and ψ_n, n = 0-∞ are orthogonal polynomials in terms of $\boldsymbol{\xi}$, it can be obtained:

$$\left\langle \Psi_i, \Psi_j \right\rangle = \left\langle \Psi_i^2 \right\rangle \delta_{ij} \text{ and } \left\langle \psi_i, \psi_j \right\rangle = \left\langle \psi_i^2 \right\rangle \delta_{ij}, \quad \forall \mathrm{i,j} = 0\text{-}\infty \tag{4}$$

where δ_{ij} is the Kronecker delta (i.e. $\delta_{ij} = 1$ if i = j and $\delta_{ij} = 0$ if $i \neq j$), $\langle \cdot,\cdot \rangle$ is the ensemble average. If f and g are two orthogonal polynomials of $\boldsymbol{\xi}$, $\langle \cdot,\cdot \rangle$ is computed by (Xiu and Karniadakis, 2002; Xiu and Karniadakis, 2003)

a. Continuous case (i.e. $\xi_{i_1}, \xi_{i_2} ... and\ \xi_{i_n}$ vary continuously over the probabilistic space):

$$\left\langle f(\boldsymbol{\xi}), g(\boldsymbol{\xi}) \right\rangle = \int\int \cdots \int f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) w(\xi_{i_1}) w(\xi_{i_2}) \cdots w(\xi_{i_n}) d\xi_{i_1} d\xi_{i_2} \cdots d\xi_{i_n} \tag{5a}$$

b. Discrete case (i.e. $\xi_{i_1}, \xi_{i_2} ... and\ \xi_{i_n}$ vary discretely over the probabilistic space):

$$\left\langle f(\boldsymbol{\xi}), g(\boldsymbol{\xi}) \right\rangle = \sum_{\xi_{i_1}} \sum_{\xi_{i_2}} \cdots \sum_{\xi_{i_n}} f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) w(\xi_{i_2}) \cdots w(\xi_{i_n}) \tag{5b}$$

where $w(\xi_{i_1})$, $w(\xi_{i_2})....w(\xi_{i_n})$ represent the weighting functions.

Equation (4) can be used to get U_i, i = 0-∞. Multiplying Eq. (2) with Ψ_i, i = 0-∞ and simplifying the resulting equations according to Eq. (5a) or (5b) give

$$U_i = \frac{\langle u \Psi_i \rangle}{\left\langle \Psi_i^2 \right\rangle}, \quad \forall i = 0\text{-}\infty \tag{6}$$

In practice, not all U_i are computed. Eq. (2) can be truncated as follows

$$u = \sum_{i=0}^{M} U_i \Psi_i(\boldsymbol{\xi}) \tag{7}$$

where M is $\frac{(n+P)!}{n!P!} - 1$, n represents the total number of uncorrelated random variables, and P is the highest order of polynomial term used to equate Ψ_i.

Tables 1-2 (e.g. Xiu and Karniadakis, 2002) list available choices of orthogonal polynomials and corresponding statistical distributions and weighting functions to generate Ψ_i, i = 0-∞, $\xi_{i_1}, \xi_{i_2} ... \xi_{i_n}$, and $w(\xi_{i_1})$, $w(\xi_{i_2})....w(\xi_{i_n})$; respectively.

Distribution	Polynomial	w(ξ)	Interval
Gaussian	Hermite polynomial $H_n(x)$	$\exp(-\xi^2)$	(-∞,∞)
gamma	Laguerre polynomial $L_n(x)$	$\exp(-\xi)$	[0, ∞]
beta	Jacobi polynomial $G_n(p, q, x)$	$(1-\xi)^{p-q}\xi^{q-1}$	[a, b]
uniform	Legendre polynomial $P_n(x)$	1	[a, b]

Table 1. Polynomials, weighting functions, and statistical distributions for generating an GPC expansion (Continuous case) (e.g. Xiu and Karniadakis, 2002)

Distribution	Polynomial	w(ξ)	Interval
Poisson	Charlier polynomial $C(x, \lambda)$	$\exp(-\lambda)\lambda^{\xi}/\xi!$	{0, 1, 2…}
binomial	Krawtchouk polynomial $K_n(x, p, N)$	$N!p^{\xi}(1-p)^{N-\xi}/[\xi!(N-\xi)!]$	{0, 1…N}
negative binomial	Meixner polynomial $M_n(x, \beta, c)$	$(\beta)_{\xi}(1-c)^{\beta}c^{\xi}/\xi!$	{0, 1, 2…}
hypergeometric	Hahn polynomial $Q_n(x, \alpha, \beta, N)$	$(\alpha+\xi)!(\beta+N-\xi)!/[\xi!\alpha!(N-\xi)!\beta!]$	{0, 1…N}

Table 2. Polynomials, weighting functions, and statistical distributions for generating an GPC expansion (Discrete case) (e.g. Xiu and Karniadakis, 2002)

Furthermore, some references (e.g. Xiu and Karniadakis, 2002; Xiu and Karniadakis, 2003) had summarized useful properties about those orthgonal polynomials listed in Tables 1-2. Interested readers may refer to these papers and these properties are not repeatedly listed here.

2.2 Discretization error estimator

Discretizing u by Eq. (7) causes some discretization errors. These discretization errors can be quantified by some error estimators. For example, the exact value of standard deviation of u (or other statistical parameters) and the one provided by Eq. (7) can be used to equate an error estimator. However, the exact value of standard deviation of u may be unavailable or difficult to be obtained. At such a situation, a Monte Carlo simulation (MCS) is required. This MCS is performed by first generating some samples of u. The standard deviation of resulting samples of u is then computed. It had been concluded (e.g. Ghanem and Spanos, 1991) that the standard deviation of u provided by an MCS will approach its exact value, if a sufficiently large amount of samples have been generated to implement the MCS. Thus, statistical parameters computed by an MCS, which is completed using a large amount of samples of u, can be used to understand the accuracy of Eq. (7).

For simplicity, this study defines two types of error estimators to quantify the discretization errors. The values of these error estimators are kept within an acceptable range when constructing a GPC representation of a random field. The first type of error estimator is equated to quantify the error ε_1 caused by truncating Eq. (2) to derive Eq. (7). At a specific $\boldsymbol{\xi}$, ε_1 is defined by (Field and Grigoriu, 2004)

$$\varepsilon_1 = \frac{1}{u_{ex}} \sum_{i=M+1}^{\infty} U_i \Psi_i = \frac{1}{u_{ex}} \left(u_{ex} - \sum_{i=0}^{M} U_i \Psi_i \right) = 1 - \frac{1}{u_{ex}} \sum_{i=0}^{M} U_i \Psi_i \tag{8}$$

where the subscript ex denotes the exact value.

The second type of error estimator is equated to quantify the errors between the exact value of standard deviation of u and the one computed based on Eq. (7). Suppose σ_{ex} and σ_{GPC} denote the exact value of standard deviation of u and the one provided by Eq. (7); respectively. The latter σ_{GPC} is computed by (Ghanem and Spanos, 1991)

$$\sigma_{GPC} = \sqrt{\sum_{i=1}^{M} U_i^2 \left\langle \Psi_i^2 \right\rangle} \tag{9}$$

Based on Eq. (9), the error ε_2 between σ_{ex} and σ_{GPC} is defined by

$$\varepsilon_2 = \frac{1}{\sigma_{ex}} \left[\sigma_{ex} - \sqrt{\sum_{i=1}^{M} U_i^2 \left\langle \Psi_i^2 \right\rangle} \right] = 1 - \frac{1}{\sigma_{ex}} \sqrt{\sum_{i=1}^{M} U_i^2 \left\langle \Psi_i^2 \right\rangle} \tag{10}$$

If σ_{ex} is unavailable or diffcult to be obtained, σ_{MCS} is substituted for it where the subscript MCS denotes the Monte Carlo simulation.

Equations (8) and (10) will be used to define the objective functions in the next section.

3. Parameter identification procedure

As stated in Sec. 1, this study considers the derivation of an GPC representation of a random field as an MO problem and applies the SPEA2 (Zitzler et al., 2001) to solve this MO

problem. Mathematically, an MO problem is defined by (Tan, et al., 2005) finding a set of vector, **P** such that

$$\underset{\boldsymbol{\Theta}\in\boldsymbol{\Phi}}{\text{Min}}\,\mathbf{F}(\boldsymbol{\Theta}) \quad \boldsymbol{\Theta}\in\Re^{N} \tag{10}$$

where $\boldsymbol{\Theta} = \{\theta_1, \theta_2 ... \theta_N\}$ is an N-dimensional vector having N parameters, $\boldsymbol{\Phi}$ defines a feasible set of $\boldsymbol{\Theta}$, and $\mathbf{F} = \{f_1, f_2 ... f_m\}$ is an objective vector with m objective functions f_i (i = 1 to m) to be minimized.

For the succeeding research, suppose the mean value and σ of a random field to be discretized have been known or computed by an MCS. It is intended to identify n and P for constructing the GPC representaton of this random field; thus, $\boldsymbol{\Theta}$ is equal to {n, P}. Meanwhile, the goals are (a) limiting the computational efforts spent in applying the resulting GPC representation; (b) keeping the accuracy of resulting GPC representation satisfying all accuracy standards. The former goal can be attained by choosing n and P within an acceptable range; while, this sudy attains the latter goal by minimizing the next three objective functions f_i, i = 1-3:

$$\begin{aligned} f_1 &= \varepsilon_1(\xi_{i_1} = \xi_{i_2} = \cdots = \xi_{i_n} = \xi_a) - S_1 \\ f_2 &= \varepsilon_1(\xi_{i_1} = \xi_{i_2} = \cdots = \xi_{i_n} = \xi_b) - S_2 \\ f_3 &= \varepsilon_2 - S_3 \end{aligned} \tag{11}$$

where ξ_a and ξ_b denote two specific values of $\boldsymbol{\xi}$ and S_i, i = 1-3 are three constants.

Based on those concepts (Tan, et al., 2005), which are frequently mentioned in solving an MO problem, Sec. 3.1 lists the general steps to identify n and P. Secs. 3.2-3.3 explains the details of specific steps.

3.1 General structure

The identification of n and P is performed by next six steps:

a. Initially, generating 2N random numbers for creating a population $\boldsymbol{\Theta}_t$ (the subscript t denotes the generation number) containing N sets of candidate values of n and P. Since n and P should be integers, each random number is rounded to its nearest integer for producing a candidate n or P value. Besides, create an empty archive $\boldsymbol{\Xi}_t$ for storing the resulting Pareto optimal set. Limit the maximum size of $\boldsymbol{\Xi}_t$ to a number N_{max}.
b. Compute the fitness value of each individual of $\boldsymbol{\Theta}_t \cup \boldsymbol{\Xi}_t$. This fitness value is defined as the sum of the number of individuals dominating an individual and the density assessment at this individual in an objective space. The density assessment is used to speed up the convergence of Pareto optimal set until it contains only nondominated individuals. Sec. 3.2 further explains the computation of fitness values of an individual.
c. Generate a temporary Pareto optimal set as follows (Tan, et al., 2005 and Zitzler, et al., 2001): Nondominated individuals of $\boldsymbol{\Theta}_t \cup \boldsymbol{\Xi}_t$ are first copied to $\boldsymbol{\Xi}_{t+1}$. If there are less than N_{max} individuals in the resulting $\boldsymbol{\Xi}_{t+1}$, sort dominated individuals of $\boldsymbol{\Theta}_t \cup \boldsymbol{\Xi}_t$ in an ascending order by their fitness values. Fill $\mathbf{Q}_{t+1}$ with first (N_{max} - $|\boldsymbol{\Xi}_{t+1}|$) dominated individuals ($|\boldsymbol{\Xi}_{t+1}|$ is the size of $\boldsymbol{\Xi}_{t+1}$). If there are more than N_{max} individuals in the resulting $\boldsymbol{\Xi}_{t+1}$, sort $\boldsymbol{\Xi}_{t+1}$ in a descending order by the distance of each individual to its k-th nearest neighbor. Then, remove last ($|\boldsymbol{\Xi}_{t+1}|$ - N_{max}) individuals from the sorted $\boldsymbol{\Xi}_{t+1}$. A reference book (Tan, et al., 2005) gives the pseudo-codes to implement this step.

d. Test whether a stopping criterion is satisfied. For example, it can stop when a maximum generation number is reached. If the stopping criterion is satisfied, the final Pareto optimal set is Ξ_{t+1}. Then, go to Step (f). Otherwise, execute the tournament selection, crossover, and mutation genetic operators to fill Θ_{t+1} with P offspring of individuals of Ξ_{t+1}. The implementation of these genetic operators is explained in Sec. 3.3.
e. Increment the generation number t. Repeat Steps (a) to (d).
f. Select manually the final values of n and P from Ξ_{t+1}.

3.2 Fitness assignment

Continuing Step (b) of Sec. 3.1, suppose the fitness value of each individual of $\Theta_t \cup \Xi_t$ to be computed. The fitness value F_i of an i-th individual is obtained by (Zitzler, et al., 2001)

$$F_i = R_i + D_i \tag{12}$$

where R_i is the raw fitness value defined based on the total number of individuals dominating the i-th individual and D_i denotes the density assessment at the i-th individual in an objective space.
The total number of individuals dominating an i-th individual is represented by a strength value S_i expressed in the form as

$$S_i = \left| \{ j \mid j \in \Theta_t \cup \Xi_t \wedge i \succ j \} \right| \tag{13}$$

where j denotes the j-th individual, $|.|$ denotes the cardinality of a set, and $\succ$ denotes the Pareto dominance. Based on Eq. (13), R_i, is defined by

$$R_i = \sum_{j \in \Theta_t \cup \Xi_t \wedge j \succ i} S_j \tag{14}$$

However, Eq. (14) may fail when most individuals do not dominate each other. Therefore, Eq. (12) includes D_i and it is computed by the distance d_i^k to its k-th nearest neighbor. The method for calculating d_i^k is as follows: First, calculate the distances of an i-th individual to all other j-th individuals of $\Theta_t \cup \Xi_t$. Thus

$$d_{ij} = \sqrt{\sum_{L=1}^{3} (f_{L,i} - f_{L,j})^2} \tag{15}$$

where the subscripts i and j denote the i-th and j-th individual; respectively. The resulting d_{ij} are then stored in a list. Sort this list and d_i^k is the k-th element of sorted list. The resulting d_i^k is used to define D_i as follows:

$$D_i = \frac{1}{d_i^k + 1} \tag{16}$$

in which 1 in the denominator is to ensure Eq. (16) is less than 1.

3.3 Genetic operators

The tournament selection, crossover, and mutation operators are applied to produce the offspring of those random numbers generated for getting candidate n and P values. Those random numbers should be coded into chromosomes before applying those genetic operators (Tan, et al., 2005).

a. n the tournament selection operation, two chromosomes are randomly selected. Discard the chromosome dominated by the other one. If else, two chromosomes are all conserved. Consider a population Ξ as an example. The tournament selection operation will repeat $|\Xi|$ times ($|\Xi|$ denotes the size of Ξ).
b. In the crossover operation, the crossover probability p_c is first defined. Two chromosomes, which have survived from the tournament selection operation, are chosen as the parents for producing the offspring. The offspring are generated by combining parts of the chromosomes contributed by each parent. Continuing using Ξ in Step (a), the crossover operation will repeat until Ξ is recovered to its original size. Figure 1(a) further illustrates this crossover operation in which chromosomes are represented by binary strings.
c. In the mutation operation, the mutation probability p_m is first set. A chromosome, which has survived from the tournament selection operation, is randomly selected. The structure of it is randomly changed to produce a new chromosome. Continuing using Ξ in Steps (a)-(b), the crossover operation will repeat $\frac{|\Xi|}{2}$ times. Figure 1(b) illustrates this mutation operator in which the chromosome to be mutated is also represented by a binary string.

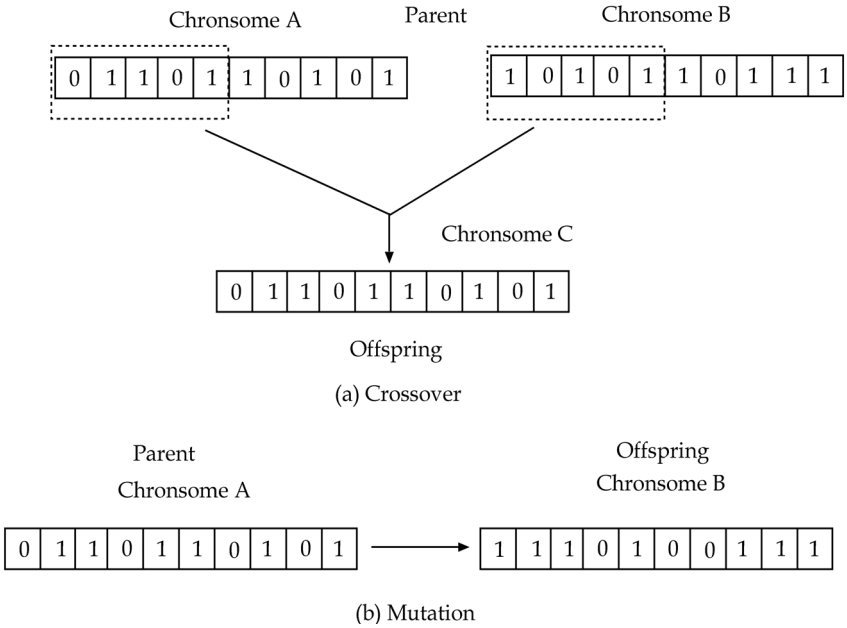


Fig. 1. Illustration of two genetic operators: (a) Crossover; (b) Mutation

4. Results

Two examples are generated to study the performance of resulting works in Sec. 3. The first example is discretizing a random field u varying with a lognormal distribution:

$$u = 1 + \log\left(0.12^2 \sum_{i=1}^{n} \xi_i\right) \tag{17}$$

where $-1 \le \xi_i$ (i = 1-2) ≤ 1 denotes n random numbers. By implementing an MCS using 10^5 samples of u, the mean μ_u and σ_u of u are calculated by 2 and 0.12; respectively. A GPC representation of u is constructed with satisfying the following accuracy standard:

$$S_j = 10\%; \quad j = 1\text{-}3 \tag{18}$$

Essential parameters for identifying n and P are set by

a. Find n and P within the range $1 \le n \le 3$ and $1 \le P \le 10$.
b. Set $\xi_a = 1$ and $\xi_b = -1.0$.
c. Set temporarily p_c and p_m are all equal to 1 and N_{max} is 100. Study subsequently the convergence of f_i, i = 1-3 with respect to different p_c and p_m values.
d. Set N = 100 in producing candidate n and P values. (That is, total 200 random numbers are generated to produce candidate n and P values). Besides, generate 100 generations of candidate n and P values.
e. Following temporarily Table 1 (e.g. Xiu and Karniadakis, 2002), apply the Hermite PC to construct an GPC representation of u. Introduce subsequently a different type of the GPC (e.g. the Legendre PC) to equate another GPC representation of u and compare the accuracy of two different GPC representations of u.

Figures 2 shows the convergence of f_i, i = 1-3. The diversity of 1st and 100th generations of random numbers for generating candidate n and P values is shown in Fig. 3. Note that less than 100 (= N) points are drawn in Fig. 2, since candidate n and P values are gotten by rounding some random numbers to their nearest integers.

Sorting the data of f_i, i = 1-3 depicted in Figs. 2-3 finds that the minization of f_3 and f_i, i = 1-2 cannot be simultaneously attained. For example, if n and P are chosen to minimize f_3 (= 0.00319), the corresponding f_i, i = 1-2 (f_1 = 0.0012; f_2 = 0.0082) are not equal to their minimum values ($f_{1,min}$ = 0.00079 and $f_{2,min}$ = 0.0001 (the subscript min denotes the minimum value)). In addition, examining Figs. 2-3 indicates that there may be two choices of final values of n and P. As listed in Table 3, if n and P are equal to 2 and 6; respectively, f_3 is minimized but f_i, i =

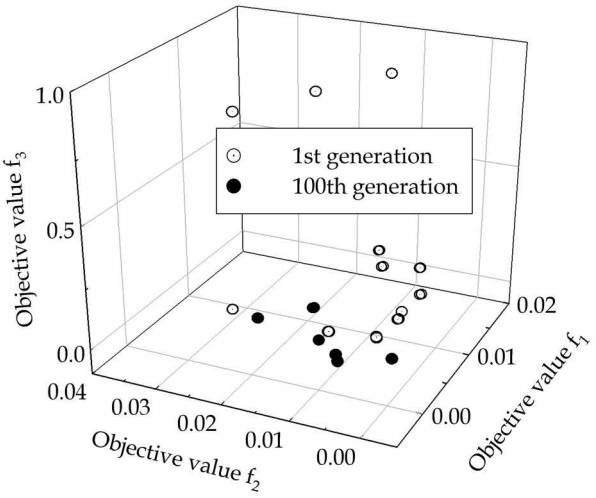


Fig. 2. Convergence of f_i, i = 1-3 (Using the Hermite PC, Lognormal distribution)

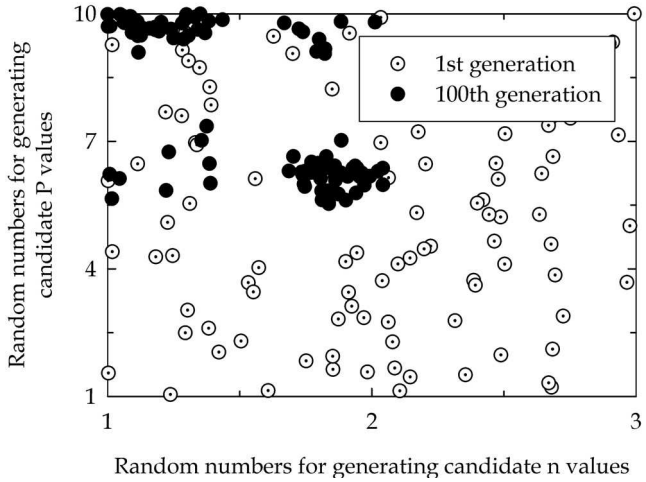


Fig. 3. Diversity of random numbers for generating candidate n and P values (Using the Hermite PC, Lognormal distribution)

1-2 are not minimized. Meanwhile, if n and P are equal to 1 and 10;respectively, f_i, i = 1-2 are minimized but f_3 is not minimized. If only one set of final values of n and P is forced to be left, this study prefers the former set. Getting a sufficiently accurate predicted standard deviation is more important.

n	P	M	f_1	f_2	f_3
2	6	28	0.0012	0.0082	0.00319
1	10	10	0.0427	0.0019	0.257

Table 3. Two different choices of final values of n and P (Using the Hermite PC, Lognormal distribution)

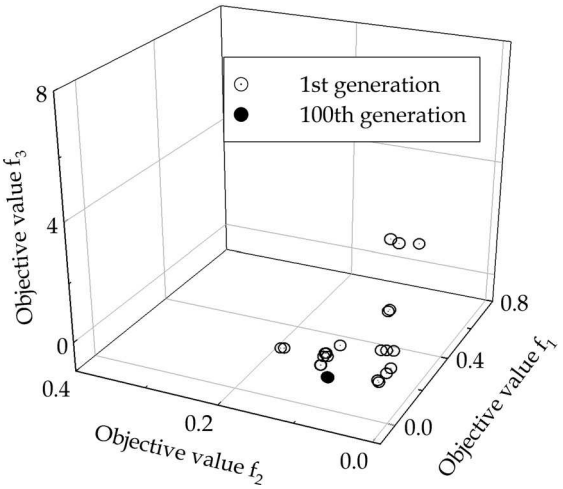


Fig. 4. Convergence of f_i, i = 1-3 (Using the Legendre PC, Lognormal distribution)

One may suspect that f_3 can be further minimized, if u is represented by another type of the GPC. As a test, Fig. 2 is modified by substituting the Legendre PC for the Hermite PC to re-discretize u. Fig. 4 depicts the convergence of resulting f_i, i = 1-3.
It seems to be difficult to compare f_3 values provided by the Legendre PC-based and Hermite PC-based representations of u by simply observing Figs. 2 and 4. Therefore, another set of n and P values, which minimize f_3, are chosen from the data for depicting Fig. 4. Table 4 lists the resulting n and P values.

n	P	M	f_1	f_2	f_3
3	2	10	0.0022	0.0778	0.08811

Table 4. Final values of n and P (Using the Legendre PC, Lognormal distribution)

Comparing Tables 3 and 4, this study suggests that a random field varying with a lognormal distribution is better represented by the Hermite PC. If the Legendre PC-based representation of u is applied, f_3 is not further minimized. In other words, the accuracy of predicted σ_u is not further improved, if the Legendre PC-based representation of u is used.
The second example is representing another random field v varying with a uniform distribution:

$$v = 1 + 0.12^2 \sum_{i=1}^{n} \xi_i \tag{19}$$

where $-1 \le \xi_i$ (i = 1-2) ≤ 1 still denote n random variables. Another MCS using 10^5 samples of v is performed. The mean value μ_v and standard deviation σ_v of v are 1 and 0.12; respectively.
Following Table 1 (e.g. Xiu and Karniadakis, 2002), the Legendre PC is applied to discretize v. With fixing other essential parameters for sketching Figs. 2-4, Fig. 5 shows the convergence of f_i, i = 1-3. The diversity of 1st and 100th generations of random numbers for generating candidate n and P values is shown in Fig. 6.

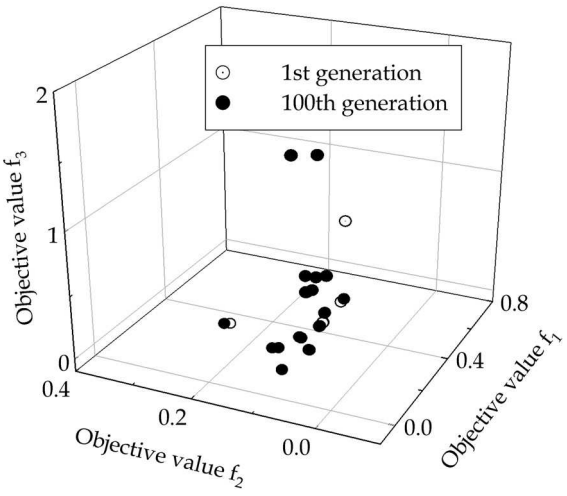


Fig. 5. Convergence of f_i, i = 1-3 (Using the Legendre PC, Uniform distribution)

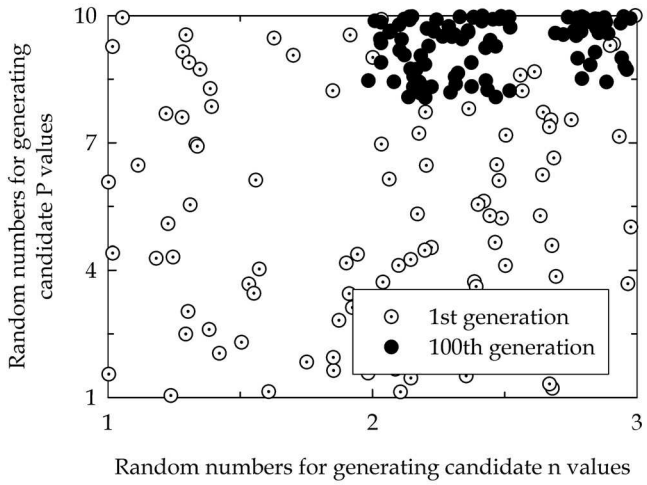


Fig. 6. Diversity of random numbers for generating candidate n and P values (Using the Legendre PC, Uniform distribution)

Sorting the data of f_i, i = 1-3 depicted in Figs. 5-6 still finds that the minization of f_3 and f_i, i = 1-2 cannot be simultaneously attained. If n and P are chosen to minimize f_3 (= 0.02635), the corresponding f_i, i = 1-2 (f_1 = 0.1363; f_2 = 0.1934) are not equal to their minimum values ($f_{1,min}$ = 0.0103 and $f_{2,min}$ = 0.0266). Also examining Figs. 5-6 finds two choices of final values of n and P. Table 5 lists these two sets of final values of n and P. If n and P are equal to 3 and 10; respectively, f_3 is minimized but f_i, i = 1-2 are not minimized. Meanwhile, if n and P are equal to 2 and 8; respectively, f_i, i = 1-2 are minimized but f_3 is not minimized. This study still prefers the former set, although more computational efforts are required in applying this set of n and P values.

n	P	M	f_1	f_2	f_3
3	10	286	0.1363	0.1934	0.0264
2	8	45	0.0103	0.0266	0.1422

Table 5. Two different choices of final values of n and P (Using the Legendre PC, Uniform distribution)

Furthermore, similarly manipulating Fig. 4, the Hermite PC is substituted for the Legendre PC to re-discretize v and the convergence of corresponding f_i, i = 1-3 is depicted in Fig. 7. Then, a set of n and P values, which minimize f_3, is chosen from the data for drawing Fig. 7. Table 6 lists the resulting n and P values.

If it is desired that the GPC representation of v should be as accurate as possible, Tables 5-6 confirms the reported conclusion (e.g. Xiu and Karniadakis, 2002) that a random field varying with a uniform distribution is better represented by the Legendre PC. Comparing Tables 5-6 indicates that the application of Hermite PC-based representation of v gives a less accurate predicted σ_v. However, if the computational efforts spent in applying a set of n and P values is the major concern, n = 2 and P = 10 may be used. The corresponding f_3 value is not too worse.

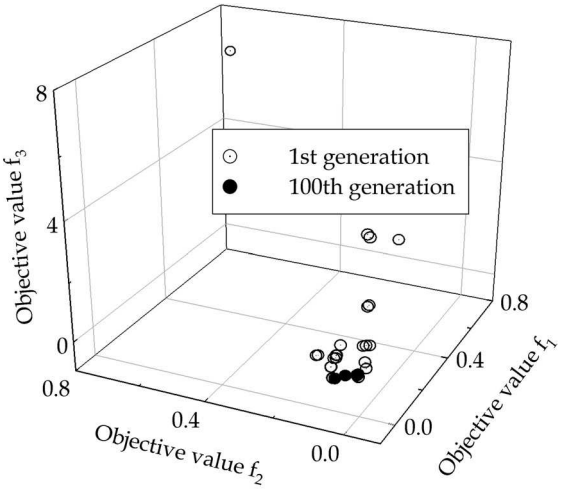


Fig. 7. Convergence of f_i, i = 1-3 (Using the Hermite PC, Uniform distribution)

n	P	M	f_1	f_2	f_3
2	10	66	0.053	0.027	0.066

Table 4. Final values of n and P (Using the Hermite PC, Uniform distribution)

Before closing this section, the effects of changing p_c and p_m on the determination of n and P values are studied. As an illustration, Figs. 3-4 are modified by changing $p_c = p_m = 1.0$ to $p_c = p_m = 0.9$. Figs. 8-9 depict the convergence of f_i, i = 1-3 and diversity of 1st and 100th generations of random numbers for producing candidate n and P values.

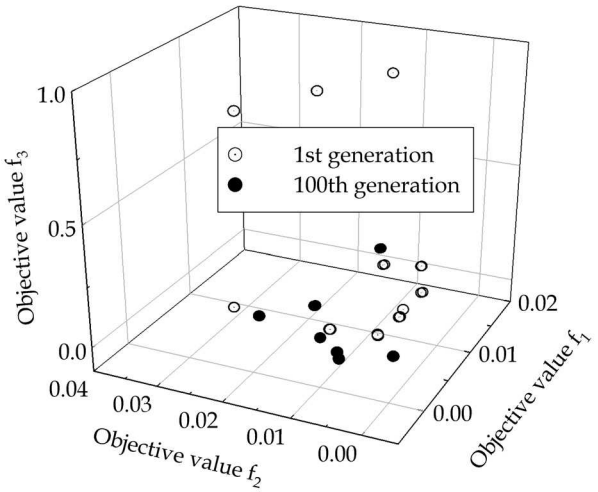


Fig. 8. Convergence of f_i, i = 1-3 with $p_c = p_m = 0.9$ (Using the Hermite PC, Lognormal distribution)

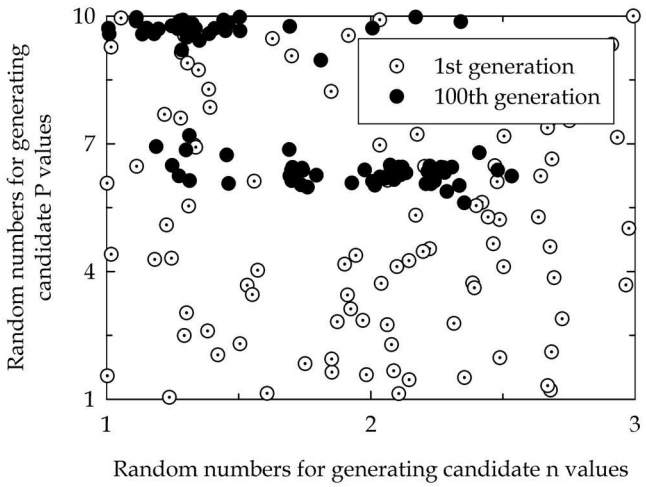


Fig. 9. Diversity of random numbers for generating candidate n and P values with $p_c = p_m = 0.9$ (Using the Hermite PC, Lognormal distribution)

Observing Figs. 8-9 finds that the decease of p_c and p_m affects insignificantly the convergence of f_i, i = 1-3 but diversifies the random numbers for generating candidate n and P values.

5. Discussion and conclusion

This study applies the SPEA2 (Zitzler, et al., 2001) to develop an auxiliary tool for identifying n and P values, which are two essential parameters for constructing a GPC representation of a random field. In Sec. 4, the proposed tool is tested to identify n and P for constructing GPC representations of two random fields varying with the lognormal and uniform distributions; respectively. The test results illustrate that an MOEA can be a good tool for constructing a sufficiently accurate GPC representation of a random field, irrespective of how many accuracy standards should be satisfied. Besides, the resulting GPC representation can be applied with keeping computational costs as few as possible.

In addition, Sec. 4 demonstrates the need of generating some pilot tests for studying the performance of GPCs in discretizing a random field before choosing one type of the GPC to discretize this random field. Such pilot tests can be quickly implemented by applying an MOEA. Tables 5-6 and Figs 5 and 7 demonstrate that the Hermite PC may be used to discretize a random field varying with a uniform distribution, if it can be sacrificed some accuracy of predicted statistical parameters (e.g. the standard deviation) of this random field.

Nevertheless, a disadvantage should be mentioned: Ψ_i, i = 0-M may be difficult to be equated in case of $n > 3$. Although much computational efforts will be spent, $n > 3$ may give more accurate GPC representation of a random field. Fortunately, consulting with some references (e.g. Field, 2004) finds that $1 \le n \le 3$ seems to be enough to generate sufficiently accurate GPC representations of random fields.

As a conclusion, an MOEA can be an efficient auxiliary tool for constructing an GPC representation of a random field satisfying multiple accuracy standards. Only Ψ_i, i = 0-M at $n > 3$ may be complementally derived in the future.

6. References

Ghanem, R. G. ; Spanos, P. D. (1991). *Stochastic Finite Elements: A Spectral Approach,* Springer-Verlag, 0387974563, New York

Xiu, D.; Karniadakis, G. E. (2003). Modelling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics,* Vol. 187, No. 1, 137-167, 0021-9991.

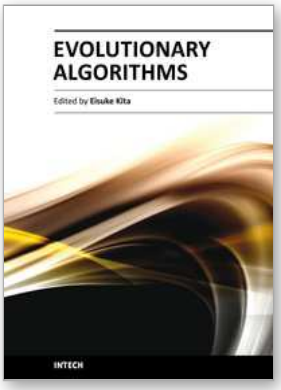
Allaix, D. L.; Carbone, V. L. (2009). Discretization of 2D random fields: A genetic algorithm approach. *Engineering Structures,* Vol. 31, Issue 5, May 2009, 1111-1119, 0141-0296.

Zitzler, E.; Laumanns, M.; Thiele, L. (2001). *SPEA2: improving the strength Pareto evolutionary algorithm.* Technical report 103. Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Switzerland.

Xiu, D.; Karniadakis, G. E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *Journal of scientific computing,* Vol. 24, No. 2, 619-664, 0885-7474.

Field, R. V. Jr.; Grigoriu, M. (2004). On the accuracy of the polynomial chaos approximation. *Probabilistic Engineering Mechanics,* Vol. 19, Issue 1-2, January 2004, 65-80, 0266-8920.

Tan, K. C., Khor, E. F., Lee, T. H. (2005). *Multiobjective evolutionary algorithms and applications.,* Springer-Verlag, 9781852338367, London, U. K.



Evolutionary Algorithms

Edited by Prof. Eisuke Kita

ISBN 978-953-307-171-8
Hard cover, 584 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Evolutionary algorithms are successively applied to wide optimization problems in the engineering, marketing, operations research, and social science, such as include scheduling, genetics, material selection, structural design and so on. Apart from mathematical optimization problems, evolutionary algorithms have also been used as an experimental framework within biological evolution and natural selection in the field of artificial life.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Guang-Yih Sheu (2011). Discretization of a Random Field – a Multiobjective Algorithm Approach, Evolutionary Algorithms, Prof. Eisuke Kita (Ed.), ISBN: 978-953-307-171-8, InTech, Available from: http://www.intechopen.com/books/evolutionary-algorithms/discretization-of-a-random-field-a-multiobjective-algorithm-approach

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821