



## A Bootstrap Study of Variance Estimation under Heteroscedasticity using Genetic Algorithm

Himadri Ghosh, *Indian Agricultural Statistics Research Institute,  
New Delhi, India. Email: hghosh@gmail.com*

M. A. Iquebal, *Indian Agricultural Statistics Research Institute,  
New Delhi, India. Email: jiqubal@gmail.com*

Prajneshu, *Indian Agricultural Statistics Research Institute,  
New Delhi, India. Email: prajneshu@yahoo.co.in*

Received: September 28, 2007    Revised: November 21, 2007

---

### Abstract

The conventional ordinary least squares (*OLS*) variance-covariance matrix estimator for a linear regression model under heteroscedastic errors is biased and inconsistent. Accordingly, several estimators have so far been proposed by various researchers. However, none of these perform well under the finite-sample situation. In this paper, the powerful optimization technique of Genetic algorithm (*GA*) is used to modify these estimators. Properties of these newly developed estimators are thoroughly studied by Monte Carlo method for various sample sizes. It is shown that *GA*-versions of the estimators are superior to corresponding *non-GA* versions as there are significant reductions in the Total relative bias as well as Total root mean square error.

*AMS Subject Classification:* 62J05, 65C05

*Key-words:* Linear regression model, Least squares estimators, Heteroscedasticity, Real-coded genetic algorithm, Bootstrap methods, Total relative bias, Total root mean square error.

---

### 1. Introduction

Linear regression models are widely used in various fields. Though these are primarily aimed at specifying the mean of a response variable as a function of explanatory variables, the nature of error variances plays an important role in the estimation and testing of regression parameters. The ordinary least squares (*OLS*) variance-covariance matrix estimator is biased and inconsistent under heteroscedastic errors (Cribari-Neto, 2004). The most commonly used heteroscedasticity-consistent covariance matrix estimator (*HCCME* or *HC*) is the one proposed by Halbert White (White, 1980). This estimator, which is also known as the *HCO* estimator, is implemented into a number of statistical software, such as *LIMDEP* and *SHAZAM*, and is commonly used by practitioners. Though White's paper continues

to have a profound impact, Monte Carlo evidence indicates that the  $HC0$  estimator can be considerably biased in finite samples and that it tends to underestimate the true variances, thus leading to the liberal associated quasi-t tests. Therefore, Cribari-Neto et al. (2000) developed a bias-correction scheme for the above estimator. Other estimators proposed are:  $HC1$  estimator (Hinkley, 1977),  $HC2$  estimator (MacKinnon and White, 1985),  $HC3$  estimator (Davidson and MacKinnon, 1993), and  $HC4$  estimator (Cribari-Neto, 2004). In terms of Error in rejection probability (ERP), the  $HC0$  estimator is outperformed by the  $HC1$  estimator, which in turn is outperformed by the  $HC2$  and  $HC3$  estimators. However, these last two estimators can not generally be ranked, although the  $HC3$  estimator is shown, in a number of Monte Carlo experiments, to be superior in typical cases. Even if the  $HC2$  and  $HC3$  estimators perform better than the  $HC0$  and  $HC1$  estimators in finite samples, their ERPs are still quite large. The  $HC4$  estimator plays an important role in testing of regression parameters but it performs poorly in estimation. All the above mentioned estimators have been studied through the resampling method, which does not guarantee any limiting property analogous to that of the true estimator. Accordingly, it makes sense to consider whether the bootstrap methods might be used to improve poor small sample performance in estimation of  $OLS$  variance-covariance matrix. To this end, Wu (1986) proposed a weighted bootstrap ( $wboot$ ) estimator of  $OLS$ , which is consistent but biased under heteroscedasticity of unknown form.

In this paper, we propose a set of new estimators for variance-covariance matrix using a recently developed optimization technique, viz. Genetic algorithm ( $GA$ ). Since  $OLS$  estimators do not take into account the heteroscedasticity aspect, a linear model under heteroscedasticity may be transformed to a standard linear model when the variance-covariance matrix is known, and to a nonlinear model, otherwise. For the latter case,  $GA$  is applied to estimate the regression and variance parameters simultaneously. After estimating the former by  $GA$ , the squared residuals are used to modify the  $HCCMEs$  and the modified ones are denoted respectively as the  $HC0-GA$ ,  $HC1-GA$ ,  $HC2-GA$ ,  $HC3-GA$ ,  $HC4-GA$  estimators. Further, a modified  $wboot$  estimator, denoted by  $wboot-GA$ , is also proposed by using the  $GA$  residuals. The proposed estimators are thoroughly studied by the Monte Carlo method for various sample sizes and their superiority over existing ones is demonstrated using the two criteria, viz. Total relative bias, and Total root mean square error (Total RMSE).

## 2. The model and estimators

The linear regression model, in which response variable ( $\mathbf{y}$ ) is modeled as a linear systematic component plus error, can be expressed as

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (2.1)$$

where  $\mathbf{y}$  is an  $n \times 1$  vector of observations,  $X$  is a fixed matrix of dimension  $n \times p$  with full column rank ( $\text{rank}(X) = p < n$ ) containing explanatory variables,  $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_p)'$  is a  $p$ -vector of unknown linear parameters, and  $\boldsymbol{\varepsilon}$  is an  $n$ -vector of errors, having mean zero and variance  $\sigma_i^2$ . We denote the covariance matrix of  $\boldsymbol{\varepsilon}$  as  $\Sigma = \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2\}$ . When the errors are homoscedastic,  $\sigma_i^2 = \sigma^2 > 0$ , i.e.  $\Sigma = \sigma^2 I_n$ , where  $I_n$  is the identity matrix of order  $n$ . The  $OLS$  estimator of  $\hat{\boldsymbol{\beta}}$  is given by  $\hat{\boldsymbol{\beta}} = (X'X)^{-1}X'\mathbf{y}$ , which has mean  $\boldsymbol{\beta}$  (i.e. it is unbiased) and variance  $\text{var}(\hat{\boldsymbol{\beta}}) = (X'X)^{-1}X'\Sigma X(X'X)^{-1} = \boldsymbol{\psi}$ , say.

The most commonly used consistent estimator for  $\psi$  is the *HCO* estimator:

$$\hat{\psi} = (X'X)^{-1}X'\hat{\Sigma}X(X'X)^{-1} \quad (2.2)$$

where  $\hat{\Sigma} = \text{diag}\{e_1^2, e_2^2, \dots, e_n^2\}$ , i.e.  $\hat{\Sigma}$  is a diagonal matrix formed out of the vector of squared *OLS* residuals. However, this estimator can be substantially negatively biased in small samples (MacKinnon and White, 1985). In order to overcome this shortcoming, several other estimators have been proposed. The *HCI* estimator uses

$$\hat{\Sigma} = n/(n-p)\text{diag}\{e_1^2, e_2^2, \dots, e_n^2\} \quad (2.3)$$

It thus includes a finite-sample correction, namely  $n/(n-p)$ , which accounts for the fact that the *OLS* residuals tend to fluctuate less than the unknown errors. The *HC2* estimator employs

$$\hat{\Sigma} = \text{diag}\{e_1^2/(1-h_1), e_2^2/(1-h_2), \dots, e_n^2/(1-h_n)\} \quad (2.4)$$

where  $h_i$  is the  $i$ th diagonal element of the 'hat matrix'  $H = X(X'X)^{-1}X'$ ,  $i = 1, 2, \dots, n$ . The quantities  $h_i$  are usually viewed as a measure of the leverage of corresponding observations.

Since statistical inference based on data resampling has drawn great attention, ordinary jackknife variance estimator for  $\hat{\beta}$  was proposed by Miller (1974). Since this estimator neglected the unbalanced nature of heteroscedastic data, Hinkley (1977) proposed a "Weighted jackknife variance estimator", which is identical with the *HCI* estimator. Further improvement towards robustness for estimating  $\text{Var}(\hat{\beta})$  was suggested by Wu (1986), which is called the "Weighted delete-one jackknife variance estimator":

$$v_{j(1)} = \sum_{i=1}^n (1-h_i)(\hat{\beta}_{(i)} - \hat{\beta})(\hat{\beta}_{(i)} - \hat{\beta})' \quad (2.5)$$

or equivalently

$$v_{j(1)} = (X'X)^{-1} \sum_{i=1}^n (1-h_i)^{-1} e_i^2 \mathbf{x}_i \mathbf{x}_i' (X'X)^{-1} \quad (2.6)$$

where  $\hat{\beta}_{(i)}$  is the *OLS* estimator of  $\beta$  with the  $i$ th observation deleted,  $e_i = y_i - \mathbf{x}_i' \hat{\beta}$  is the  $i$ th residual and  $h_i = \mathbf{x}_i'(X'X)^{-1}\mathbf{x}_i$ ,  $\mathbf{x}_i'$  is the  $i$ th row of  $X$ . Wu (1986) showed that

$$E(e_i^2) = (1-h_i)\sigma_i^2 + O(n^{-1}),$$

which leads to

$$E(v_{j(1)}) = \text{Var}(\hat{\beta})\{1 + O(n^{-1})\} \quad (2.7)$$

Hoaglin and Welsh (1978) used the diagonal elements  $h_1, h_2, \dots, h_n$  of the hat matrix  $H = X(X'X)^{-1}X'$  as a measure of leverage of  $n$  observations, since  $h_i = \partial \hat{y}_i / \partial y_i$ , where  $\hat{y}_i$  is the  $i$ th fitted value. As noted by Davidson and MacKinnon (1993):

$$\hat{\beta}_{(i)} = \hat{\beta} - \{1/(1-h_i)\}(X'X)^{-1}\mathbf{x}_i' e_i, \quad i = 1, 2, \dots, n \quad (2.8)$$

It follows that, when  $e_i$  is large and/or  $(1-h_i)$  is small (i.e.,  $h_i$  is large), effect of the  $i$ th

observation on at least some of the elements of  $\hat{\boldsymbol{\beta}}$  is likely to be sizable. We can also write

$$\mathbf{x}'_i \hat{\boldsymbol{\beta}}_{(i)} = \mathbf{x}'_i \hat{\boldsymbol{\beta}} - \{h_i/(1-h_i)\}e_i, \quad i = 1, 2, \dots, n \quad (2.9)$$

thus implying that the change in the  $i$ th fitted value caused by omission of observation  $i$  equals  $-e_i h_i/(1-h_i)$ . As a direct consequence, change in the  $i$ th residual is  $\{h_i/(1-h_i)\}e_i$ . Hence, it is quite meaningful to obtain an estimator of variance of  $\hat{\boldsymbol{\beta}}$  through jackknifing in such a way that the *OLS* estimator obtained by deleting high leverage point will get a low weight as in eq. (2.5). Accordingly, an alternative estimator, widely known as the *HC3* estimator, with superior finite-sample behavior, was defined by modifying  $\hat{\Sigma}$  appearing in eq. (2.2) as

$$\hat{\Sigma} = \text{diag}\{e_1^2/(1-h_1)^2, e_2^2/(1-h_2)^2, \dots, e_n^2/(1-h_n)^2\} \quad (2.10)$$

It discounts the squared *OLS* residuals more heavily and provides a close approximation to the jackknife estimator considered by MacKinnon and White (1985). The finite-sample corrections included in the definition of the *HC2* and *HC3* estimators are therefore based on the degrees of leverage of different observations.

Another estimator proposed by Cribari-Neto (2004), known as the *HC4* estimator, uses

$$\hat{\Sigma} = \text{diag}\{e_1^2/(1-h_1)^{\delta_1}, e_2^2/(1-h_2)^{\delta_2}, \dots, e_n^2/(1-h_n)^{\delta_n}\} \quad (2.11)$$

where  $\delta_i = \min\{4, h_i/\bar{h}\}$ , and  $\bar{h} = n^{-1} \sum_{i=1}^n h_i$ . The exponent controls the level of discounting for observation  $i$  and is given by the ratio between  $h_i$  and the average of  $h_i$ 's. Since  $\delta_i > 0$ , it follows that  $0 < (1-h_i)^{\delta_i} < 1$ . Hence, the  $i$ th squared *OLS* residual is more strongly inflated, when  $h_i$  is large relative to  $\bar{h}$ . However, when  $h_i/\bar{h} > 4$  (i.e.  $h_i > 4p/n$ ),  $\delta_i = 4$ ; thus maximum amount of discounting is restricted to 4.

### 3. Genetic algorithm optimization procedure

Monte Carlo evidence shows that the *HCCMEs* described in previous section tend to display poor finite-sample behavior in terms of the downward bias when the design matrix  $X$  contains high leverage points. As mentioned in the Introduction section, the resultant model is nonlinear when variance-covariance matrix is unknown. Generally, Levenberg-Marquardt's nonlinear estimation procedure is used to obtain the parameter estimates. This is an iterative method and requires initial values. The procedure is repeated till convergence takes place. Nevertheless, it may converge to a local optimum. This problem may be avoided by using *GA*, which is a Stochastic search and optimization procedure motivated by the principles of genetics and natural selection. Some fundamental ideas of genetics are borrowed and used artificially to construct the search algorithms that are robust and require minimal problem information. The working principle of *GA* is very different from that of the classical optimization techniques. We first discuss the binary-coded *GA* in short and then shift our attention to real-coded *GA*, which is used in the present study. The first step of *GA* is to encode the parameter values into an appropriate range with finite-length digital strings (i.e. chromosomes, usually binary strings of length  $\alpha$ ). A widely used formula for decoding is

$$c = L + \{A/(2^B - 1)\} \times (U - L) \quad (3.1)$$

where  $c$  is the encoded value of a chromosome,  $U$  and  $L$  are the upper and lower bounds of the parameter to be estimated, and  $A$  and  $B$  are respectively the decoded value and number of digits of the chromosome.

The three operators, viz. Selection, crossover, and mutation make *GA* an important tool for optimization. When a string or a parameter solution is created by *GA*, it is evaluated in terms of its fitness, which is the Residual sum of squares (*RSS*). Selection operator of *GA* is performed to identify good solutions, to make its multiple copies and to eliminate bad solutions from the population. There exists a number of ways to achieve the above tasks. Some common methods are: Tournament selection, Proportionate selection, and Ranking selection (Deb, 2002). The most common method of selection that has better convergence and computational time efficiency is the Tournament selection. Since selection operator can not create a new solution, the crossover and mutation operators are used in mating pool to create a new population. There exists a number of crossover operators in the *GA* literature. Different string pairs are expected to have some good bit representation. This is because only  $\alpha$  pairs of different string representations are possible to be created and the pairs of strings used in crossover have already survived selection pressure. If bad solutions are created, they get eliminated in the next selection operation and hence have a short life. The crossover operator should not be allowed to use all strings in a population in order to preserve some good strings selected during the selection operator. If a crossover probability  $p_c$  is used,  $100 p_c\%$  strings in the population are used in the crossover and rest of the population is simply copied to the new population. The mutation operator is used to ensure diversity in the population. It alters a string locally to create a better string (parameter solution). If certain genes (digits) of all chromosomes (strings) in the population are identical, their values will never change after selection and crossover. This will reduce the chance for some new chromosomes to enter this population, thereby lending the *GA* procedure into the trap of local optima. To avoid this situation, mutation with very small probability, say 0.01 is required.

When the binary-coded *GA* is used to handle problems having a continuous search space, a number of difficulties arise. One of these is that of Hamming cliffs, which causes artificial hindrance to a gradual search. The other difficulty is its inability to achieve any arbitrary precision in arriving at the optimal solution. Thus, the crossover operator used in binary coding needs to be redesigned in order to increase the propagation of more meaningful schemata pertaining to a continuous search space. Since real parameters are used directly (without any string coding), solving a real-parameter optimization problem is easier as compared to one with a binary-coded *GA*. To use the selection operator in real-coded *GA*, the Tournament selection is generally employed. Here, tournaments are played between two solutions and the better solution is chosen and placed in the mating pool. The process is continued and the set of solutions with improved fitness value is obtained in the next generation. A pair of real-parameter decision variable vector is used to create a new pair of offspring vectors and a decision variable vector is perturbed to a mutated vector in a meaningful manner, as discussed below.

### 3.1. Crossover operator

Search power of a crossover operator is defined as a measure of flexibility of the operator to create a pair of offspring in the search space. The design of crossover operator is characterized by the spread factor  $\beta$  (Deb, 2002) as

$$\beta = |(c_1 - c_2)/(p_1 - p_2)| \quad (3.2)$$

where  $c_i$ 's and  $p_i$ 's are the offspring and parent points respectively,  $i = 1, 2$ . Crossover operators having spread factor  $\beta < 1$ ,  $\beta = 1$  and  $\beta > 1$  are called the Contracting, Stationary and Expanding crossovers respectively. Specifically, we first consider those two parent strings of length  $\alpha$ , which are crossed at a site  $k \in (0, \alpha - 1)$  from the rightmost bit. Represent two parent strings as  $a_i, b_i \in (0, 1)$ , whose decoded values are:

$$x_1 = \sum_{i=0}^{\alpha-1} a_i 2^i, \quad x_2 = \sum_{i=0}^{\alpha-1} b_i 2^i$$

Using the property of single-point crossover operator, decoded values of offspring strings become

$$y_1 = \sum_{i=0}^{k-1} b_i 2^i + \sum_{i=k}^{\alpha-1} a_i 2^i, \quad y_2 = \sum_{i=0}^{k-1} a_i 2^i + \sum_{i=k}^{\alpha-1} b_i 2^i$$

It is easy to show that the mean of parent point is same as that of the offspring point. Thus, we can write the spread factor  $\beta$  as

$$\beta(k) = \{1 - \xi(k)\} / \{1 + \xi(k)\}$$

where

$$\xi(k) = \sum_{i=0}^{k-1} u_i 2^i / \sum_{i=k}^{\alpha-1} u_i 2^i, \quad u_i = (a_i - b_i), \quad \text{for } i = 0, 1, \dots, (\alpha - 1).$$

Substituting  $u_i = 1$ , we obtain the following expression for  $\beta$ :

$$\beta(k) = 1 - 2(2^k - 1)/(2^k - 1) \approx 1 - 2^{(-\alpha+1)} 2^k \quad (3.3)$$

For a particular value of  $\beta$ , the number of cross-sites that will produce spread factor  $\beta$  in the range  $(\beta, \beta + d\beta)$  is obtained by calculating the slope  $dk = 1/|d\beta/dk|_\beta d\beta$ . The frequency of cross-sites may be considered as a probability distribution of  $\beta$  in the limiting case as  $d\beta \rightarrow 0$ . Hence, by differentiating eq. (3.3) with respect to  $k$  and substituting the expression for  $k$  in terms of  $\beta$ , we obtain

$$C(\beta) = \omega / (1 - \beta) \quad (3.4)$$

where  $C(\beta)$  is the probability of occurrence of  $\beta$ , and  $\omega$  is a constant. The functional form of the above distribution for  $0 \leq \beta \leq 1$  is shown in Figure 1. As  $\beta$  is increased towards one, probability increases and this is true for the other values of  $u_i$ ,  $i = 0, 1, 2, \dots, (\alpha - 1)$ . For each contracting crossover with spread factor  $\beta$ , there exists an expanding crossover with the spread factor  $1/\beta$  obtained by treating an offspring as parent and vice-versa. Thus for expanding crossover, i.e.  $\beta > 1$ , we obtain the probability of occurrence of spread factor between  $\beta$  and  $(\beta + d\beta)$  the same as that of the contracting crossover with the spread factor

between  $1/(\beta + d\beta)$  to  $1/\beta$ . Therefore

$$\varepsilon(\beta)[(\beta + d\beta) - \beta] = C(1/\beta)[1/\beta - 1/(\beta + d\beta)]$$

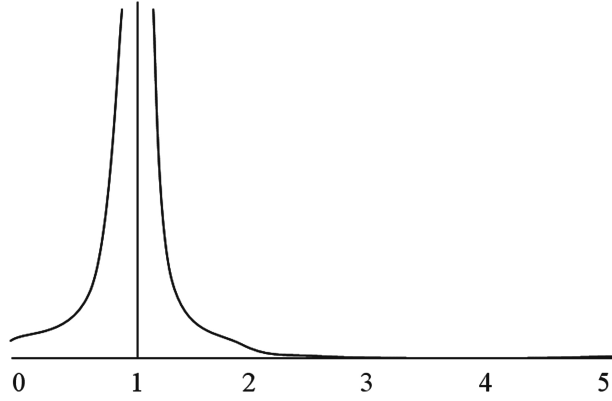
Taking limits as  $d\beta \rightarrow 0$ , a straightforward algebra yields

$$\varepsilon(\beta) = C(1/\beta)/\beta^2 \tag{3.5}$$

Thus, the probability density function for the expanding crossover  $\beta$  is:

$$\varepsilon(\beta) = \omega/\{\beta(\beta - 1)\} \tag{3.6}$$

which is also plotted in Figure 1. As the probability of creating the offspring strings having a large  $\beta$  is small, the probability of creating offspring points closer to parent points is larger than that of the points far away from parents.



**Figure 1.** Probability density function of contracting and expanding crossovers

In order to simulate the crossover operator of a single-point binary crossover directly on real variables, a crossover operator that uses a similar probability distribution, called Simulated binary crossover (*SBX*) operator, was developed by K. Deb in 1995. It works with two parent solutions and creates two offspring. The procedure of computing the offspring  $x_i^{(1,t+1)}$  and  $x_i^{(2,t+1)}$  from parent solutions  $x_i^{(1,t)}$  and  $x_i^{(2,t)}$ , where  $x_i^{(j,t)}$  is the value of  $i$ th variable of  $j$ th parent in  $t$ th generation,  $i = 1, 2, \dots, p$ ;  $j = 1, 2$ ; and  $t \geq 1$  is described below. By drawing a rectangular random variable over  $[0, 1]$  and using

$$\beta_{q_i} = \begin{cases} (2u_i)^{-(\eta_c+1)} & \text{if } u_i \leq 0.5 \\ \{2(1 - u_i)\}^{\eta_c+1}, & \text{otherwise} \end{cases} \tag{3.7}$$

the probability distribution of the spread factor  $\beta_i$  in  $i$ th variable, similar to that in the single-point crossover in binary-coded *GA*, is obtained. Specifically, the probability density function of  $\beta_i$  is (Deb and Agrawal, 1995):

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \leq 1 \\ 0.5(\eta_c + 1)(1/\beta_i)^{\eta_c+2}, & \text{otherwise} \end{cases} \tag{3.8}$$

where  $\eta_c$  is a nonnegative real number. These are better approximations for  $C(\beta)$  and  $\varepsilon(\beta)$ ,

when  $\eta_c$  is large, which gives higher probability for creating ‘near-parent’ solutions, while its small value allows distant solutions to be selected as an offspring. So, a moderate value of  $\eta_c$ , say in the range 2 to 5, is chosen. After obtaining  $\beta_{q_i}$  from eq.(3.8), the offspring are calculated as follows:

$$\begin{aligned} x_i^{(1,t+1)} &= 0.5\{(1 + \beta_{q_i})x_i^{(1,t)} + (1 - \beta_{q_i})x_i^{(2,t)}\} \\ x_i^{(2,t+1)} &= 0.5\{(1 - \beta_{q_i})x_i^{(1,t)} + (1 + \beta_{q_i})x_i^{(2,t)}\} \end{aligned} \quad (3.9)$$

It may be noted that  $\beta_{q_i}$  satisfies the relation

$$x_i^{(2,t+1)} - x_i^{(1,t+1)} = \beta_{q_i}\{x_i^{(2,t)} - x_i^{(1,t)}\} \quad (3.10)$$

Eq. (3.9) implies that the two offspring are symmetrically distributed and the points of symmetry are equi-spaced from the mid-point of parent solutions. Thus, biasedness towards any particular parent solution is avoided.

### 3.2. Mutation operator

The Mutation operator does the same task as performed by the real parameter crossover operator. The only difference between the two is that the former perturbs every parent parameter solution to create a new population while the latter perturbs two parent parameter solutions at a time to produce two new offspring. With only one parent, a range of perturbation must be predefined. There are several mutation operators, like Polynomial, Random, and Non-uniform. However, one advantage of the Polynomial mutation operator is that the probability distribution does not change with the generations, thereby avoiding local optima. Therefore, in our study, we shall be employing this operator. It mutates the  $i$ th variable  $x_i^{(1,t+1)}$  to  $y_i^{(1,t+1)}$  by the transformation:

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + \{x_i^{(U)} - x_i^{(L)}\} \bar{\delta}_i \quad (3.11)$$

where  $\bar{\delta}_i$  follows the Polynomial probability distribution

$$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}, |\delta| \leq 1, \text{ and } \eta_m \geq 0$$

The parameter  $\eta_m$  is responsible for controlling the shape of probability distribution of  $y_i^{(1,t+1)}$ .

### 4. Development of new estimators using genetic algorithm

In the presence of heteroscedasticity, if we ignore the effect of estimation of the variance-covariance matrix of *OLS* through conventional statistic, it will lead to an invalid inference of the regression parameters. All attempts made so far to improve the finite-sample reliability of the *HC* estimators considered use of leverage-based modifications. An advantage of *GA* technique is that it is capable of estimating the regression and variance parameters simultaneously. The residuals obtained through *GA* are expected to estimate  $\sigma_i^2$  with re-



duced bias as compared to those obtained through *OLS* residuals. Therefore, we propose a modification of existing estimators by using squared *OLS* residuals obtained through *GA* technique and study their finite-sample properties. The modified forms for estimator of  $\Sigma$ , viz. *HCO-GA*, *HC1-GA*, *HC2-GA*, *HC3-GA*, *HC4-GA* are respectively as follows:

$$\begin{aligned}\hat{\Sigma}_{GA} &= \text{diag}\{e_{GA1}^2, e_{GA2}^2, \dots, e_{GAN}^2\}, \\ \hat{\Sigma}_{GA} &= n/(n-p)\text{diag}\{e_{GA1}^2, e_{GA2}^2, \dots, e_{GAN}^2\}, \\ \hat{\Sigma}_{GA} &= \text{diag}\{e_{GA1}^2/(1-h_1), e_{GA2}^2/(1-h_2), \dots, e_{GAN}^2/(1-h_n)\}, \\ \hat{\Sigma}_{GA} &= \text{diag}\{e_{GA1}^2/(1-h_1)^2, e_{GA2}^2/(1-h_2)^2, \dots, e_{GAN}^2/(1-h_n)^2\},\end{aligned}$$

and

$$\hat{\Sigma}_{GA} = \text{diag}\{e_{GA1}^2/(1-h_1)^{\delta_1}, e_{GA2}^2/(1-h_2)^{\delta_2}, \dots, e_{GAN}^2/(1-h_n)^{\delta_n}\},$$

where  $e_{GAi}$ ,  $i = 1, 2, \dots, n$  are the residuals obtained after fitting the nonlinear model through *GA*.

## 5. Comparison of estimators through bootstrap simulation

Numerical results show that the estimators based on asymptotic *HCCMEs* can be seriously misleading for small samples (Flachaire, 2005). Instead of recomputing the point estimator by deleting observation(s) each time, bootstrap estimators are considered due to their limiting property analogous to that of the true estimator. Two methods are considered here for studying the performance of estimators of *OLS* variance-covariance matrix under heteroscedasticity for regression model (2.1). In the first method discussed below, performance of the *HCCMEs* is studied based on drawing i.i.d. samples  $\{u_i^*\}_1^n$  from the normalized residuals  $\{e_i/(1-pn^{-1})^{1/2}\}_1^n$ , where  $e_i = y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}$  is the *i*th *OLS* residual. Define the *i*th bootstrap observation  $y_i^* = \mathbf{x}_i' \hat{\boldsymbol{\beta}} + u_i^*$  by treating  $\hat{\boldsymbol{\beta}}$  as “true” parameter and  $\{e_i/(1-pn^{-1})^{1/2}\}$  as the “population” of residuals. From each bootstrap replication, the *HCCMEs* are computed.

We now study the performance of newly proposed estimators by modifying the above mentioned bootstrap resampling method through *GA*. In this case,  $\hat{\boldsymbol{\beta}}$  and  $\hat{\mathbf{u}}_{GA}$  are treated as “true” parameter and “population” of residuals respectively, where  $\hat{u}_{GAi} = \{e_{GAi}/(1-pn^{-1})^{1/2}\}$  and  $e_{GAi} = (y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_{GA})$  is *i*th residual obtained through fitting of nonlinear model using *GA*. A large number, say *B* of bootstrap samples each of size *n* are constructed by drawing i.i.d. samples  $\{u_{GAi}^*\}_1^n$  from the population of *GA* residuals  $\hat{u}_{GAi}$  and  $y_i^*$ , the *i*th response variable is calculated using  $y_i^* = \mathbf{x}_i' \hat{\boldsymbol{\beta}} + u_{GAi}^*$ . From each bootstrap replication, residuals are obtained through *GA*. These *GA* residuals are used to compute different estimators, viz. *HCO-GA*, *HC1-GA*, *HC2-GA*, *HC3-GA* and *HC4-GA* for  $\text{Var}(\hat{\boldsymbol{\beta}})$ . This method, however, does not adapt well to the possibility of heteroscedasticity.

The second method of resampling residuals to construct the bootstrap estimator, proposed by Wu (1986), which is consistent for both homoscedasticity and heteroscedasticity

of unknown form, is as follows:

- (i) A master data set is generated using the heteroscedastic model. Now, the model is fitted using *OLS* method to get regression parameter estimator  $\hat{\boldsymbol{\beta}}$  and residual  $\mathbf{e}$
- (ii) For each  $i, i = 1, 2, \dots, n$ , draw a random number  $t_i^*$  from a population with mean zero and variance one
- (iii) Construct a bootstrap sample  $(y^*, X)$ , where  $y_i^* = \mathbf{x}_i' \hat{\boldsymbol{\beta}} + t_i^* e_i / \sqrt{(1-h_i)}$ ,  $\mathbf{x}_i'$  denoting the  $i$ th row of  $X$
- (iv) Compute *OLS* estimator of  $\boldsymbol{\beta}$  as  $\hat{\boldsymbol{\beta}}^* = (X'X)^{-1} X' \mathbf{y}^*$
- (v) Another bootstrap sample  $(\mathbf{y}^{**}, X)$  is constructed, where  $y_i^{**} = \mathbf{x}_i' \hat{\boldsymbol{\beta}}^* + t_i^{**} e_i^* / \sqrt{(1-h_i)}$ , and  $e_i^*$  is the  $i$ th residual generated from fitted model to bootstrap sample in Step (iii)
- (vi) Variance of  $\hat{\boldsymbol{\beta}}^{**} = (X'X)^{-1} X' \mathbf{y}^{**}$  is computed as

$$v^* = (X'X)^{-1} \sum_{i=1}^n (e_i^*)^2 / (1-h_i) \mathbf{x}_i \mathbf{x}_i' (X'X)^{-1}$$

- (vii) Repeat Steps (iii) to (vi) a large number of times, say,  $B$ .

In respect of regression model given in Step (v), the variance-covariance matrix of  $\hat{\boldsymbol{\beta}}^{**}$  given in Step (vi) is known as the “Weighted bootstrap (*wboot*) estimator”. The above bootstrap method is modified using the residuals obtained from *GA* to study the quantum of reductions of the Total relative bias and Total RMSE for  $\text{Var}(\hat{\boldsymbol{\beta}})$ . Thus, Steps (iii) and (v)-(vi) are respectively modified on replacing  $e_i$  by  $e_{GAi}$ , and  $e_i^*$  by  $e_{GAi}^*$ , where  $e_{GAi}$ 's and  $e_{GAi}^*$  are respectively the *GA* residuals generated from fitted model to master data and bootstrap sample. Accordingly, in respect of the modified regression model given in modified Step (v), variance-covariance matrix of  $\hat{\boldsymbol{\beta}}^{**}$  given in Step (vi) is known as the “Weighted bootstrap-*GA* (*wboot-GA*) estimator”.

The numerical results are obtained using the model:

$$y_i = \beta_1 + \beta_2 x_i + \sigma_i \varepsilon_i, \quad i = 1, 2, \dots, n \quad (5.1)$$

To obtain the heteroscedastic data, we consider

$$\sigma_i^2 = \exp\{\gamma x_i + \gamma x_i^2\}$$

for different values of  $\gamma$ , viz.  $\gamma = 0.0, 0.04, 0.08, 0.12$ . The degree of heteroscedasticity can be expressed using  $\lambda = (\text{maximum } \sigma_i^2) / (\text{minimum } \sigma_i^2)$ , where  $\lambda = 1$  under homoscedasticity and  $\lambda > 1$ , otherwise. The sample sizes considered are  $n = 50, 100, 150$ . When  $n = 50$ , the values of  $x_i$  are independent random draws from a lognormal distribution. These observations are each replicated twice and thrice respectively, when  $n = 100$  and  $150$ . The errors  $\varepsilon_i$  follow i.i.d  $N(0, 1)$  distribution. Using  $\beta_1 = 1$ , and  $\beta_2 = 0$ , the master data set is generated. The number of bootstrap replications is 999. The Total relative bias is defined as the sum of absolute values of individual relative biases of estimated variances of  $\beta_1$  and  $\beta_2$ , where the individual relative bias is the difference between means of all variances and true variance divided by the true variance. Further, the Total RMSE is defined as the square root of sum of mean square errors of  $\beta_1$  and  $\beta_2$ . Using the above mentioned bootstrap methods, the Total relative bias and Total RMSE for the existing estimators, viz. *HC0, HC1, HC2, HC3, HC4*

and *wboot* estimators are computed. As there is hardly any difference in the results obtained in respect of the *HC1* and *HC2* estimators, results for the former, in subsequent discussion, are omitted to save space.

The Total relative bias using *OLS* residuals for the *HC0*, *HC2*, *HC3*, *HC4* and *wboot* estimators are computed and reported in Table 1. A perusal shows that it generally decreases as the sample size increases, but for a fixed sample size, it increases as the heteroscedasticity increases. For instance, when  $n = 50$  and  $\gamma = 0.12$ , the Total relative bias of the *HC0*, *HC2*, *HC3*, *HC4*, and *wboot* estimators are 0.58, 1.44, 0.36, 1.88, and 0.32 respectively. But with increase in the sample size (from 50 to 150) and the same level of heteroscedasticity, the Total relative bias reduces to 0.21, 0.97, 0.15, 1.00, and 0.14 respectively. However, its rate of increase corresponding to various levels of heteroscedasticity decreases as the sample size increases. For instance, the Total relative bias for the *HC3* estimator increases from 0.21 to 0.36, 0.15 to 0.24, and 0.09 to 0.15, when  $n = 50, 100$ , and 150 respectively. The *HC4* estimator provides most biased results, followed respectively by the *HC2* and *HC0* estimators. Further, the Total relative bias is minimum in the case of the *HC3* and *wboot* estimators, except for the case of homoscedasticity (i.e.  $\gamma = 0.0$ ) for the *HC0* estimator. For instance, when  $n = 50$ , and  $\gamma = 0.08$ , the Total relative bias for the *HC0*, *HC2*, *HC3*, *HC4*, and *wboot* estimators are 0.47, 1.40, 0.36, 1.88, and 0.32 respectively.

**Table 1.** Total relative bias (using residuals obtained through *OLS*)

$n$	$\gamma$	$\lambda$	<i>HC0</i>	<i>HC2</i>	<i>HC3</i>	<i>HC4</i>	<i>wboot</i>
50	0.00	1.00	0.1936	0.6125	0.2123	0.6783	0.2195
	0.04	4.57	0.3255	0.9065	0.2866	1.1881	0.2725
	0.08	20.86	0.4739	1.4044	0.3146	1.6238	0.3119
	0.12	95.27	0.5867	1.4434	0.3610	1.8884	0.3217
100	0.00	1.00	0.1083	0.4319	0.1571	0.4071	0.1073
	0.04	4.57	0.2472	0.9123	0.1967	0.9534	0.1744
	0.08	20.86	0.2922	0.9816	0.2249	1.0274	0.1741
	0.12	95.27	0.3177	1.1349	0.2465	1.2033	0.2311
150	0.00	1.00	0.0877	0.3977	0.0987	0.3001	0.0916
	0.04	4.57	0.1114	0.5916	0.1079	0.6492	0.1038
	0.08	20.86	0.1954	0.7083	0.1437	0.7178	0.1347
	0.12	95.27	0.2166	0.9706	0.1573	1.0017	0.1469

We now describe the procedure for computation of the Total relative bias in respect of GA-versions of the above estimators. The model used for fitting through the Real-coded GA is

$$y_{1i} = x_{1i}\beta_1 + x_{2i}\beta_2 + \varepsilon_i, \quad i = 1, 2, \dots, n \tag{5.2}$$

where  $y_{1i} = y_i/\sigma_i$ ,  $x_{1i} = 1/\sigma_i$ ,  $x_{2i} = x_i/\sigma_i$ . Now, the transformed model is nonlinear in the parameters. Using GA, we can compute the estimators for regression parameters  $\beta$  and variance parameter  $\gamma$  simultaneously, which in turn is used to obtain  $\hat{\sigma}_i$ . During the GA

process, the objective (fitness) function, which is to be minimized, is:

$$\sum_{i=1}^n (y_{1i} - x_{1i}\beta_1 - x_{2i}\beta_2)^2 \quad (5.3)$$

The real-coded GA with SBX operator for  $\eta_c = 2$  is applied for simulation study. The GA parameters, viz. population size, crossover probability, and mutation probability for minimization of eq.(5.3) are taken as 50, 0.9, 0.01 respectively and the number of generations is 100. The GA terminated with an accuracy level of  $10^{-3}$  in generally 9 out of 10 runs. This shows that the GA methodology is successful in fitting the linear regression models with heteroscedastic errors.

By drawing a large number, say  $B = 999$  of i.i.d. samples each of size  $n$  ( $= 50, 100$  and  $150$ ) from “population” of GA residuals  $\hat{u}_{GAi}$ , 999 bootstrap samples are constructed. From each bootstrap replication using the real-coded GA, the residuals are computed to obtain various estimators, viz. *HC0-GA*, *HC1-GA*, *HC2-GA*, *HC3-GA*, and *HC4-GA* for  $\text{Var}(\hat{\beta})$ . Also, by performing modified weighted bootstrap using residuals obtained through GA, a set of *wboot-GA* variance estimators is obtained. The Total relative bias for each estimator is computed and presented in Table 2. It may be pointed out that the GA-versions of all the estimators perform much better than their corresponding non-GA counterparts. For example, the Total relative bias is 33% less for the *wboot-GA* estimator as compared to that of the *wboot* estimator. The simulation study shows that the improvement through the GA-versions is more marked in the presence of heteroscedasticity. Further, the *wboot-GA* estimator generally performs best followed respectively by the *HC3-GA* and *HC0-GA* estimators, except for some homoscedastic situations. It may be noted that almost same amount of reduction in the Total relative bias by the *wboot* estimator can be achieved by using the GA-version of the *HC0* estimator. The added advantage of the latter is that it is computationally much easier than the former.

**Table 2.** Total relative bias (using residuals obtained through GA)

$n$	$\gamma$	$\lambda$	<i>HC0-GA</i>	<i>HC2-GA</i>	<i>HC3-GA</i>	<i>HC4-GA</i>	<i>wboot-GA</i>
50	0.00	1.00	0.1883	0.4318	0.2191	0.4378	0.2157
	0.04	4.57	0.2253	0.5905	0.1830	0.5786	0.1524
	0.08	20.86	0.2754	0.6617	0.2077	0.6498	0.1859
	0.12	95.27	0.4155	0.8192	0.2634	0.8034	0.2279
100	0.00	1.00	0.1793	0.3851	0.1809	0.3119	0.1782
	0.04	4.57	0.1491	0.4788	0.1384	0.4769	0.1195
	0.08	20.86	0.1974	0.5997	0.1437	0.5734	0.1248
	0.12	95.27	0.2123	0.6139	0.1739	0.6267	0.1466
150	0.00	1.00	0.0952	0.3151	0.0962	0.3037	0.0936
	0.04	4.57	0.1048	0.3747	0.1031	0.3945	0.1015
	0.08	20.86	0.1156	0.4987	0.1072	0.5152	0.1061
	0.12	95.27	0.1284	0.5535	0.1089	0.5338	0.1073

The Total RMSE (using residuals obtained through *OLS*) for various variance estimators are presented in Table 3. It may be pointed out that the *HC0* estimator has smallest Total RMSE followed by that of the *wboot* estimator while the *HC2* and *HC4* estimators generally perform poorly. Table 4 depicts the Total RMSE for the GA-versions of various estimators. A perusal shows that these provide a good improvement over the corresponding conventional estimators. For instance, when  $n = 150$  and  $\gamma = 0.12$ , the Total RMSE for the *HC0*, *HC2*, *HC3*, *HC4* and the *wboot* estimators are 0.20, 0.32, 0.24, 0.28 and 0.25 respectively, which are decreased to 0.17, 0.26, 0.18, 0.24 and 0.20 respectively for the *HC0-GA*, *HC2-GA*, *HC3-GA*, *HC4-GA* and *wboot-GA* estimators. It is noticed from Table 4 that the *HC0-GA* estimator generally performs best followed by the *wboot-GA* estimator. Further, from Tables 3 and 4, the *HC0-GA* estimator is found to perform better than the *wboot* estimator.

**Table 3.** Total RMSE (using residuals obtained through *OLS*)

$n$	$\gamma$	$\lambda$	<i>HC0</i>	<i>HC2</i>	<i>HC3</i>	<i>HC4</i>	<i>wboot</i>
50	0.00	1.00	0.0135	0.0312	0.0166	0.0285	0.0168
	0.04	4.57	0.0382	0.0915	0.0572	0.0901	0.0559
	0.08	20.86	0.1319	0.4482	0.2319	0.3969	0.2264
	0.12	95.27	0.6991	1.9612	0.8464	1.8754	0.9282
100	0.00	1.00	0.0049	0.0142	0.0051	0.0193	0.0052
	0.04	4.57	0.0111	0.0533	0.0146	0.0347	0.0139
	0.08	20.86	0.1016	0.2478	0.1149	0.2217	0.1195
	0.12	95.27	0.4184	0.8605	0.4852	0.8083	0.5257
150	0.00	1.00	0.0033	0.0093	0.0039	0.0068	0.0051
	0.04	4.57	0.0136	0.0327	0.0149	0.0327	0.0153
	0.08	20.86	0.0984	0.1944	0.1073	0.1158	0.1107
	0.12	95.27	0.1966	0.3197	0.2379	0.2819	0.2464

**Table 4.** Total RMSE (using residuals obtained through *GA*)

$n$	$\gamma$	$\lambda$	<i>HC0</i>	<i>HC2</i>	<i>HC3</i>	<i>HC4</i>	<i>wboot</i>
50	0.00	1.00	0.0064	0.0107	0.0088	0.0126	0.0082
	0.04	4.57	0.0151	0.0499	0.0369	0.0883	0.0139
	0.08	20.86	0.1909	0.4125	0.1855	0.3863	0.1927
	0.12	95.27	0.7129	0.9393	0.7618	0.9157	0.6242
100	0.00	1.00	0.0041	0.0095	0.0047	0.0097	0.0043
	0.04	4.57	0.0074	0.0281	0.0013	0.0141	0.0125
	0.08	20.86	0.0909	0.1951	0.1066	0.2065	0.1019
	0.12	95.27	0.3824	0.8246	0.3369	0.7962	0.3233
150	0.00	1.00	0.0027	0.0063	0.0028	0.0052	0.0024
	0.04	4.57	0.0067	0.0141	0.0068	0.0173	0.0069
	0.08	20.86	0.0813	0.1758	0.0905	0.1078	0.0934
	0.12	95.27	0.1722	0.2613	0.1814	0.2409	0.2031

To sum up, it may be concluded from the above study that the GA-versions of the *wboot* and *HCO* estimators perform best when both the criteria, viz. Total relative bias and Total RMSE are taken into consideration. Therefore, if existence of high-power computational facilities is not a limitation, the *wboot-GA* estimator may be employed for the purpose. Otherwise, priority should be given to the *HCO-GA* estimator over the *wboot* estimator. This clearly brings out the important role played by Genetic algorithm in the Linear regression analysis for estimation of *OLS* variance-covariance matrix with unknown form of heteroscedasticity.

## 6. Conclusion

The importance of recently developed powerful optimization technique of Genetic algorithm is highlighted to develop GA-based estimators of *OLS* variance-covariance matrix for the linear regression model under heteroscedasticity. In finite sample study, our simulation results show that the proposed estimators perform better than the corresponding existing estimators in terms of their capability to reduce the Total relative bias and the Total root mean square error. It is recommended that either the *wboot-GA* estimator or the *HCO-GA* estimator may be employed for estimation of the *OLS* variance-covariance matrix with unknown form of heteroscedasticity. This would enable efficient computation of the standard errors of *OLS* of regression parameters in a real-life data analysis. The estimators of the standard errors may, in turn, be used to carry out interval estimation of the regression parameters.

As a future research work, performance of the proposed GA-based estimators of variance-covariance matrix of *OLS* estimator is being currently studied for testing of regression parameters by quasi-t test. In this regard, bootstrap critical value would be computed for testing the null hypothesis regarding significance of regression parameter by using the GA residuals obtained from the restricted model. Performance of the critical value would be studied through bootstrap procedure for various sample sizes and levels of heteroscedasticity. In the case of bootstrap variance estimation, the null rejection probability would be estimated through the bootstrap *p*-value. The error in rejection probability would be examined by using this value along with the asymptotic critical value. Finally, power function of the test statistic based on the newly developed estimators of variance-covariance matrix of *OLS* estimator would be computed. Some progress on the above aspects is already made and shall be reported separately in due course of time.

## Acknowledgements

The authors are grateful to the referees for their valuable comments.

## References

- Cribari-Neto, F., 2004. Asymptotic inference under heteroscedasticity of unknown form. *Computational Statistics and Data Analysis*, 45, 215–233.
- Cribari-Neto, F., Ferrari, S. L. P., Cordeiro, G. M., 2000. Improved heteroscedasticity-consistent variance estimators. *Biometrika*, 87, 907–918.

- Davidson, R., MacKinnon, J. G., 1993. *Estimation and Inference in Econometrics*. Oxford University Press, New York.
- Deb, K., Agrawal, R. B., 1995. Simulated binary crossover for continuous search space. *Complex Systems*, 9, 115–148.
- Deb, K., 2002. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Pvt. Ltd., Singapore.
- Flachaire, E., 2005. Bootstrapping heteroscedastic regression models: Wild bootstrap vs pair bootstrap. *Computational Statistics and Data Analysis*, 49, 361–376.
- Hinkley, D.V., 1977. Jackknifing in unbalanced situations. *Technometrics*, 19, 285–292.
- Hoaglin, D. C., Welsch R. E., 1978. The hat matrix in regression and ANOVA. *The American Statistician*, 32, 17–22.
- MacKinnon, J. G., White H., 1985. Some heteroscedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29, 305–325.
- Miller, R. G., 1974. Unbalanced jackknife. *Annals of Statistics*, 2, 880–891.
- White, H., 1980. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48, 817–838.
- Wu, C. F. J., 1986. Jackknife bootstrap and other resampling methods in regression analysis. *Annals of Statistics*, 14, 1261–1295.