# Orthogonal Least Trimmed Absolute Deviation Estimator for Multiple Linear Errors-in-Variables Model

Huirong Cao
Langfang Teachers College
College of Mathematics and Information Science
Langfang, Hebei 065000
P. R. CHINA
Correspondence author: huirongcao@126.com

Fuchang Wang
Institute of Disaster Prevention
Basic Courses Department
Sanhe, Hebei 065201
P. R. CHINA
fzmath@126.com

*Abstract:* Orthogonal least trimmed absolute deviation (OLTAD) estimator of the multiple linear errors-in-variables (EIV) model is presented. We show that the OLTAD estimator has the high breakdown point and appropriate properties. A new decimal-integer-coded genetic algorithm(DICGA) and Fast-OLTAD method for solving OLTAD estimators are also proposed. Computational experiments of the OLTAD estimator of the multiple linear EIV model on benchmark data and synthetic data are provided. The results indicate that the DICGA and Fast-OLTAD methods perform well in dealing with high leverage outliers in reasonable computational time.

*Key–Words:* Linear errors-in-variables model, Robust estimator, Orthogonal least trimmed absolute deviation estimator, Decimal-integer-coded genetic algorithm, Fast-OLTAD method, Outliers

## 1 Introduction

The multiple errors-in-variables (EIV) model is just the regression model with both dependent and independent variables being subject to error. Studies on multiple EIV model aroused many investigators' interest in recent decades. The sensitivity of linear EIV model to outliers has been noticed, and robust methods have been developed. The work includes the S [2]and M estimator [3],the robust PCA method [4], weighted orthogonal regression method [5], least trimmed squares estimator [6], fast and robust estimation [7], t-type estimator [8] for EIV model. However, these robust methods are too complicated, especially in computation. Simple robust method is needed so that it can be applied to more practical problems.

In this paper, we propose a new and simple robust method for multiple linear EIV model, which is called orthogonal least trimmed deviation(OLTAD) estimator. In section 2, we give the definition, properties and computing complexity analysis of OLTAD estimator for multiple linear EIV model . In section 3, we give a basic introduction to genetic algorithm and develop an algorithm by using decimal-integer-coded genetic algorithm(DICGA) that is computationally feasible for large size samples to compute the OLTAD estimator for the multiple EIV model. Since the objective function of the OLTAD estimator is continuous, non-differentiable and non-convex, showing multiple local minima, so the computation of the OLTAD estimator is laborious according to the direct optimization method. In fact, although Jung (2007) proposed a fast algorithm for the the orthogonal least trimmed squares(OLTS) estimator in a multiple EIV regression, our OLTAD estimator is easier to be computed than the OLTS estimator. The reason is detailed in section 2. In addition, we also propose a Fast-OLTAD algorithm for computing the OLTAD estimator. In the final sections 4 and 5, simulations and a numerical example are given to illustrate the effectiveness of the OLTAD estimator. From a numerical example, we observed that the OLTAD estimator gives very useful influence information about observations in multiple EIV model.

## 2 The Orthogonal Least Trimmed Absolute Deviation Estimator of Multiple Linear EIV Model

### 2.1 The Definition of OLTAD Estimator

In statistics, EIV models are regression models that account for measurement errors in the independent variables. In contrast, standard regression models assume that those regressors have been measured exactly, or observed without error,i.e. those models account only for errors in the dependent variables, or responses.

When the independent variables have been measured with errors, estimators based on the standard assumption leads to inconsistent estimates, meaning that the parameter estimates do not tend to the true values even in very large samples. For simple linear regression the effect is an underestimate of the coefficient, known as the attenuation bias. In non-linear EIV models, the direction of the bias is likely to be more complicated.

In this paper, we consider the multiple linear EIV model as follows

$$\left\{ \begin{array}{lcl} \boldsymbol{y}_i & = & [1\ \boldsymbol{x}_i^T]\boldsymbol{\beta} + \varepsilon_i \\ \mathbf{X}_i & = & \boldsymbol{x}_i + \boldsymbol{u}_i \end{array} \right. \quad (i = 1, 2, \cdots, n.) \qquad (1)$$

where $\boldsymbol{\beta} = [\beta_0, \beta_1, \cdots, \beta_p]^T \in \mathbb{R}^{p+1}$, $\beta_0$ is an intercept term, $(\mathbf{X}_i^T, y_i)^T$, $i = 1, 2, \cdots, n.$ are observations, The $p$ dimensional vectors $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T$ are unknown design points and $(\varepsilon_i, \boldsymbol{u}_i^T), i = 1, 2, \cdots, n$ are independently distributed model statistical errors.

For estimated regression coefficients $\hat{\beta}_1, \cdots, \hat{\beta}_p$ and an estimated intercept $\hat{\beta}_0$, we denote the orthogonal residuals

$$r_i(\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_p) = \frac{y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip})}{\sqrt{1 + \hat{\beta}_1^2 + \hat{\beta}_2^2 + \cdots + \hat{\beta}_p^2}}$$

Further, let $u_{i:n}$ denote the $i$-th order statistic of $n$ numbers $u_1, \cdots, u_n$. The high-breakdown estimator we consider is defined as follows:

**Definition 1** *Let $r_i$ be the i-th orthogonal residual determined by a multiple linear EIV model with parameters $\hat{\beta}_0, \cdots, \hat{\beta}_p$ and a given data set $\mathbb{Z}$. The orthogonal least trimmed absolute deviation (OLTAD) estimator is given by*

$$\textit{OLTAD}(\mathbb{Z}) = \arg\min_{\hat{\beta}_0, \cdots, \hat{\beta}_p} \sum_{i=1}^{h_p} \{|r_1|, \cdots, |r_n|\}_{i:n} \qquad (2)$$

*where $h_p$ with $[n/2] \leqslant h_p \leqslant n$ is a parameter influencing the estimation.*

Let us consider the global robustness of the OLTAD estimator. This is the finite-sample version of breakdown point, introduced by Donoho & Huber . The breakdown point of an estimator $T(W)$ with a sample $W$ is defined as

$$\epsilon^*(T) = \min\{\frac{m}{n} | \sup_{\tilde{W}} \|T(W) - T(\tilde{W})\| = \infty\}$$

where $\tilde{W}$ is obtained by replacing $m$ observations by arbitrary points.

Roughly speaking, the breakdown point is the smallest fraction of the contaminated data to make the estimator useless. The breakdown point of the least trimmed squares(LTS) method equals about 1/2 as the sample size $n$ goes to the infinity. Rousseeuw & Leroy (Chapter 3) [18].

The OLTAD estimator is an $h$-sample estimator, so the following proposition is evident. Taking $h \approx n/2$ the OLTAD estimator has a 50% breakdown points, it means that the OLTAD estimator is also a high breakdown estimator – see also Rousseeuw & Leroy (Chapter 3) [18]. Although both OLTS and OLTAD estimators are high breakdown estimator, OLTAD estimator computation needn't C-Step unlike OLTS estimator. So, the computation time of OLTAD estimator is fewer, i.e., the computation of OLTAD estimator is faster.

### 2.2 Necessary Conditions for the Optimal Solution of Orthogonal Least Absolute Deviation Estimator

For the convenience of the discussion, we adopt different samples data character in this part. We discuss the problems in $\mathbb{R}^p$, and draw the conclusions. In the next subsection, these conclusion will be used for the basement of the new algorithms.

Let $G = \{\boldsymbol{x}_i | \boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T \in \mathbb{R}^p, i = 1, 2, \cdots, m\}$ is data consisting of $m$ samples in $\mathbb{R}^p$, $w_i$ is weight of the $i$-th sample.

We search a hyperplane $\alpha^T \boldsymbol{x} + \beta = 0$, with $\|\alpha\| = 1$ in $\mathbb{R}^p$, such that

$$S(\alpha, \beta) = \sum_{i=1}^{m} w_i \left|\alpha^T \boldsymbol{x}_i + \beta\right| \qquad (3)$$

getting minimum value .

To ensure the uniqueness of the fitting hyperplanes, we assume that $rank([\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]) \geqslant p$. If $rank([\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]) = p$, then the only optimal hyperplane is determined by $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$. Next, the proof is under the following condition $rank([\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]) > p$.

**Theorem 2** *(Existence of Optimal Hyperplane) There exists $\hat{\boldsymbol{\alpha}}$, $\hat{\beta}$ such that*

$$S(\hat{\alpha}, \hat{\beta}) = \min_{\alpha, \beta} \sum_{i=1}^{m} w_i |\boldsymbol{\alpha}^T \boldsymbol{x}_i + \beta|.$$

**Proof:** There exists hypersphere

$$S_r = \{\boldsymbol{x} | \boldsymbol{x}^T \boldsymbol{x} = r^2, r > 0\} \qquad (4)$$

in $\mathbb{R}^p$, makes $G \subset U(O, r)$.

Because if exists optimal hyperplane, it must be overlapped with hypersphere, so it only need check wether existing optimal hyperplane or not in them.

$\forall x_0 \in S_r$, then $\boldsymbol{x}_0^T \boldsymbol{x}_0 = r^2$, we make tangent hyperplane $\boldsymbol{x}_0^T \boldsymbol{x} = r^2$ through the point $\boldsymbol{x}_0$ in the hypersphere, shift this tangent hyperplane and a series of tangent hyperplanes $\boldsymbol{x}_0^T \boldsymbol{x} + b = (-r^2 \leqslant b \leqslant r^2)$ are obtained. Thus, all tangent hyperplanes through the hypersphere are $hp : \boldsymbol{x}_0^T \boldsymbol{x} + b = 0, -r^2 \leqslant b \leqslant r^2, \boldsymbol{x}_0 \in S_r$, and the weights sums from the points in $G$ to hyperplane $hp$ is

$$\begin{aligned} S(\boldsymbol{\alpha}, \beta) = \ & S(\frac{x_0^T}{\sqrt{x_0^T \boldsymbol{x}_0}}, \frac{b}{\sqrt{\boldsymbol{x}_0^T \boldsymbol{x}_0}}) \\ = \ & \sum_{i=1}^{m} w_i \frac{\left|\boldsymbol{x}_0^T \boldsymbol{x}_i + b\right|}{\sqrt{\boldsymbol{x}_0^T x_0}} \end{aligned} \quad (5)$$

Because the definition domain $S_r \times [-r, r]$ of $S(\alpha, \beta)$ is a bounded and closed set in $\mathbb{R}^{p+1}$, and $S(\alpha, \beta)$ is continuous function on $S_r \times [-r, r]$,$S(\boldsymbol{\alpha}, \beta)$ can obtain the minimum on $S_r \times [-r, r]$ based on multiple continuous function's properties, i.e. there exists $\hat{\boldsymbol{\alpha}}, \beta$ make $S(\boldsymbol{\alpha}, \beta)$ obtain the minimum.

**Theorem 3** *In $\mathbb{R}^p$, if the optimal hyperplane $\hat{\boldsymbol{\alpha}}^T \boldsymbol{x} + \hat{\beta} = 0$ pass through p − 1 points in G, then it at least pass through another point in G.*

**Proof:** Here,we use reduction to absurdity method to prove this theorem.

In $\mathbb{R}^p$, if the optimal hyperplane

$$\hat{\boldsymbol{\alpha}}^T \boldsymbol{x} + \hat{\beta} = 0$$

only pass through $p - 1$ points in $G$, then we may assume that the optimal hyperplane equation is $\boldsymbol{x}_p = 0$, otherwise, the optimal hyperplane equation can be reduced to $\boldsymbol{x}_p = 0$ by include translate and rotate transformations.

Set the points in the optimal hyperplane $G$ as follows:

$$\begin{aligned} &(x_{11}, x_{12}, \cdots, x_{1p-1}, 0), (x_{21}, x_{22}, \cdots, x_{2p-1}, 0), \\ &\qquad \cdots, (x_{p1}, x_{p2}, \cdots, x_{pp-1}, 0) \end{aligned} \quad (6)$$

Because a $p - 1$ dimension hyperplane can be determined by $p$ points, the $p$ points can be placed on the $p - 1$ dimension hyperplane by coordinate transform methods

$$\begin{cases} x_p = 0, \\ x_1 = 0. \end{cases}$$

and

$$\begin{aligned} \mathbf{x}_1 &= (0, x_{12}, \cdots, x_{1p-1}, 0), \\ \boldsymbol{x}_2 &= (0, x_{22}, \cdots, x_{2p-1}, 0), \\ &\cdots, \\ \boldsymbol{x}_p &= (0, x_{p2}, \cdots, x_{pp-1}, 0) \end{aligned} \quad (7)$$

We may assume that the points above and below the hyperplane $x_p = 0$ are $x_{p+1}, x_{p+2}, \cdots, x_t$ and $x_{t+1}, x_{t+2}, \cdots, x_m$ in $G$ respectively. When $p + 1 \leqslant i \leqslant t$, $x_{ip} > 0$, while $t + 1 \leqslant i \leqslant m$, $x_{ip} < 0$. The sum of orthogonal distances from points in $G$ to optimal hyperplane is

$$\begin{aligned} S((0, 0, \cdots, 1), 0) &= \sum_{i=1}^{m} w_i \left|x_{ip}\right| \\ &= \sum_{i=p+1}^{t} w_i x_{ip} - \sum_{i=t+1}^{m} w_i x_{ip} \end{aligned} \quad (8)$$

Let hyperplane $\mathbf{x_p} = \tan\theta \cdot \mathbf{x_1}$ in $\mathbb{R}^p$, the $x_1, \mathbf{x}_2, \cdots, \mathbf{x}_p$ are in the hyperplane $\mathbf{x_p} = \tan\theta \cdot \mathbf{x_1}$. Hence, there exists $\delta > 0$, $\forall\theta \in (-\delta, \delta)$, the points $\boldsymbol{x}_{p+1}, \boldsymbol{x}_{p+2}, \cdots, \boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2}, \cdots, \boldsymbol{x}_m$ still stay at the two side of the plane $\boldsymbol{x}_p = \tan\theta \cdot \boldsymbol{x}_1$, i.e. when $p + 1 \leqslant i \leqslant t$, $\boldsymbol{x}_{ip} > \tan\theta \cdot \boldsymbol{x}_{i1}$ and when $t + 1 \leqslant i \leqslant m$, $\boldsymbol{x}_{ip} < \tan\theta \cdot \boldsymbol{x}_{i1}$. Now, the sum of weights orthogonal distances from the points of $G$ to the optimal hyperplane $\boldsymbol{x}_p = \tan\theta \cdot \boldsymbol{x}_1$, i.e. $\tan\theta \cdot \boldsymbol{x}_1 - \boldsymbol{x}_p = 0$ is

$$\begin{aligned} & S((\tan\theta, 0, \cdots, 0, -1), 0) \\ &= \sum_{i=1}^{m} w_i \frac{\left|\tan\theta \cdot \boldsymbol{x}_{i1} - \boldsymbol{x}_{ip}\right|}{\sqrt{1 + \tan^2\theta}} \\ &= \frac{1}{\sqrt{1 + \tan^2\theta}} [\sum_{i=p+1}^{t} w_i (x_{ip} - \tan\theta \cdot x_{i1}) - \\ &\quad \sum_{i=t+1}^{m} w_i (x_{ip} - \tan\theta \cdot x_{i1})] \\ &= \cos\theta [\sum_{i=p+1}^{t} w_i (x_{ip} - \tan\theta \cdot x_{i1}) \\ &\quad - \sum_{i=t+1}^{m} w_i (x_{ip} - \tan\theta \cdot x_{i1})] \\ &= \sum_{i=p+1}^{t} w_i (\cos\theta \cdot x_{ip} - \sin\theta \cdot x_{i1}) \\ &\quad - \sum_{i=t+1}^{m} w_i (\cos\theta \cdot x_{ip} - \sin\theta \cdot x_{i1}) \\ &= \cos\theta (\sum_{i=p+1}^{t} w_i \boldsymbol{x}_{ip} - \sum_{i=t+1}^{m} w_i x_{ip}) \\ &\quad - \sin\theta (\sum_{i=p+1}^{t} w_i x_{i1} - \sum_{i=t+1}^{m} w_i x_{i1}) \end{aligned} \quad (9)$$

Since $\theta = 0$, $x_p = 0$ is the optimal hyperplane, thus

$$\begin{aligned}&[S((\tan\theta,0,\cdots,0,-1),0)]'_\theta\,|_{\theta=0}\\ &=[-\sin\theta(\sum_{i=p+1}^{t} w_i x_{ip}-\sum_{i=t+1}^{m} w_i x_{ip})\\ &-\cos\theta(\sum_{i=p+1}^{t} w_i \boldsymbol{x}_{i1}-\sum_{i=t+1}^{m} w_i x_{i1})]|_{\theta=0}\\ &=-(\sum_{i=p+1}^{t} w_i x_{i1}-\sum_{i=t+1}^{m} w_i x_{i1})=0\end{aligned} \tag{10}$$

Thus,when $\forall\ \theta \in \overset{\circ}{U}(0,\delta)$, there is

$$\begin{aligned}&S((\tan\theta,0,\cdots,0,-1),0)\\ &=\cos\theta(\sum_{i=p+1}^{t} w_i x_{ip}-\sum_{i=t+1}^{m} w_i x_{ip})\\ &<\sum_{i=p+1}^{t} w_i x_{ip}-\sum_{i=t+1}^{m} w_i x_{ip}\\ &=S((0,0,\cdots,1),0)\end{aligned} \tag{11}$$

this result is inconsistent with that $\boldsymbol{x}_p = 0$ is the optimal hyperplane. So, this theorem is proved.

**Theorem 4** *(necessary conditions 1) The optimal hyperplane* $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ *in* $\mathbb{R}^p$ *at least pass through* $p$ *points in* $G$.

**Proof.** (1) If the optimal hyperplane $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ pass through $p-1$ points, Then by theorem 2, it should pass through $p$ points in $G$.
(2) If the optimal hyperplane $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ only pass through $t$ , $0 \leqslant t < p-1$ points in $G$ , then we pick the other $p-1-t$ different points in the optimal hyperplane which were labeled as $\boldsymbol{x}_{m+1}, \boldsymbol{x}_{m+2}, \cdots, \boldsymbol{x}_{m+p-1}$ and in the $m+p-1$ points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m$, $\boldsymbol{x}_{m+1}, \boldsymbol{x}_{m+2}, \cdots, \boldsymbol{x}_{m+p-1}$, the hyperplane $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ is still optimal and pass through $p-1$ points in $m+p-1$ points, it at least pass through another point belonging to $G$ in the $m+p-1$ points set,thus, it at least pass through $k+1$ points in $G$, the results are inconsistent with each other. This finishes the proof.

**Theorem 5** *(necessary conditions 2) If the optimal hyperplane* $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ *in* $\mathbb{R}^p$ *pass through* $t(p \leqslant t < m)$ *points* $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_t}$ *in* $G$*, then rank of vector group* $\boldsymbol{x}_{i_2} - \boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_3} - \boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_t} - \boldsymbol{x}_{i_1}$ *is* $r = p-1$

**Proof.** If the rank of vector group $r < p-1$, then they can make a $r$ dimension hyperplane, the points $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_t}$ lie in the $r$ dimension hyperplane. So, these points can be placed on the $p-2(r \leqslant p-2)$ dimension hyperplanes $\begin{cases}\mathbf{x_p} = \mathbf{0}\\ \boldsymbol{x}_1 = 0\end{cases}$. From the process of theorem 2, $\hat{\boldsymbol{\alpha}}^T\boldsymbol{x} + \hat{\beta} = 0$ is not the optimal hyperplane, they are inconsistent with each other. So, this theorem is proved.

From Theorem 4 and Theorem 5, we can find that when $p < m$, the optimal hyperplane at least pass through the $p$ points $\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_t}$ in $G$ , and the rank of the vector group $\boldsymbol{x}_{i_2} - \boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_3} - \boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_t} - \boldsymbol{x}_{i_1}$ is $p-1$.

Thus, if fitting the optimal hyperplane of $G = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$ by the orthogonal least deviation criterion, it only requires being searched in the hyperplanes determining by $p$ points in $G$. The number of $p-1$-dimension hyperplanes is $\binom{m}{p}$ at most and the number of the optimal hyperplanes is also $\binom{m}{p}$ at most.

For example, for four vertices of regular tetrahedron, its each face is the optimum plane. For fixed $p$, with $m$ increasing, the computational complexity of the algorithm is $O(m^p)$.

## 2.3 Properties of OLTAD Estimator

Assume also that any $p \times p$ submatrix of $\boldsymbol{X}$ is nonsingular; in this case we say that observations are in general position. Then the finite sample breakdown point of the OLTAD estimator in the regression model (2) satisfies $\epsilon^*(\hat{\boldsymbol{\beta}}_{OLTAD}) = ([(n-p)/2]+1)/n$. Consequently it has a 50% asymptotic breakdown point.

Cui(1997) proved strong consistency and asymptotic normality of orthogonal least absolute deviation(OLAD) estimator [20] in linear EIV regression model. However, his proof is long and complicated. Thus, we will do theoretical research on the strong consistency and asymptotic normality of OLTAD estimator in future.

In general, random sampling technique and bootstrap method, such as [21] and [22]. Therefore, in this place, we can use a jackknife estimator for the asymptotic covariance matrix of the slope vector $\hat{\boldsymbol{\beta}}_{OLTAD}$. The jackknife estimator can be written by

$$\begin{aligned}\frac{1}{n-1}\sum_{i=1}^{n}&(\hat{\boldsymbol{\beta}}_{OLTAD,(i)} - \hat{\boldsymbol{\beta}}_{OLTAD,(\bullet)})\\ &(\hat{\boldsymbol{\beta}}_{OLTAD,(i)} - \hat{\boldsymbol{\beta}}_{OLTAD,(\bullet)})^T\end{aligned} \tag{12}$$

where $\hat{\boldsymbol{\beta}}_{OLTAD,(i)}$ is the OLTAD estimator with the $i$-th observation omitted and $\hat{\boldsymbol{\beta}}_{OLTAD,(\bullet)} = \frac{1}{n}\sum_{i=1}^{n}\hat{\boldsymbol{\beta}}_{OLTAD,(i)}$.

### 2.4 Computing Complexity Analysis of OLTAD Estimator

From the definition of OLTAD estimator, we can find that if we compute all the least absolute deviation estimators based upon all possible $\binom{n}{h}$ subset from data set $\mathbb{Z}$, compare all the $\underset{\hat{\beta}_0,\cdots,\hat{\beta}_p}{\arg}\sum_{i=1}^{h}\{|r_1|,\cdots,|r_n|\}_{i:n}$, then the best fitting estimator corresponding the minimum of all $\underset{\hat{\beta}_0,\cdots,\hat{\beta}_p}{\arg}\sum_{i=1}^{h}\{|r_1|,\cdots,|r_n|\}_{i:n}$ always exists.

Unfortunately, the computation of these estimators is quite hard. More precisely, the best fitting estimator with the exact fit property are NP-hard to compute depending on $n$ and $h$. When accepting the widely believed assumption that the complexity classes NP and P are not equal [10], we therefore have no hope to compute exact solutions for large high dimensional data sets. Therefore, the computation of these estimators is a very challenging and interesting problem.

It is well known that the LTS estimator is given by the least squares estimator for the $h$-subset whose sum of squared residuals is a minimum. In the linear EIV regression model, the exact algorithm is a full search algorithm for all $\binom{n}{h}$ subsets in the orthogonal regression model. Practically, this algorithm works if the number of observations is less than 30. Some approximate algorithms have been proposed. One of them is the PROGRESS algorithm by Rousseeuw & Leroy [18] which is a resampling algorithm. Hawkins D.M [16] introduced a feasible set algorithm for the LTS estimate that in a random $h$-subset it exchanges one observation with another observation if such exchange decreases the value of criterion.

Rousseeuw and Van Driessen [17] show that C-Step procedure guarantees an improvement in the objective value of the LTS estimator, Jung [6] modifies the C-Step procedure by replacing vertical residuals by orthogonal residuals, and shows that new C-Step procedure also can guarantee an improvement in the objective value of the OLTS estimator. The Fast-OLTS algorithm for OLTS estimator also is proposed by Jung [6].

One of the practical attractions of our OLTAD estimator is the relative ease (compared with the OLTS estimator) which it can be computed. Because Feng [19] proves that an important property, just like least absolute deviation(LAD) estimator of linear model, the OLTAD estimator corresponds to an exact fit to some subset of size $p+1$. So the OLTAD estimator is similarly characterized as a two-part problem, i.e., we identify the correct subset of size $h$ to cover with the OLTAD criterion firstly, and subsequently determine the subset of size $p+1$ that minimizing the sum of orthogonal absolute deviations to these $h$ cases which the number of subsets of size $h$ from a sample of size $n$. There are $\binom{n}{p+1}$ elemental subsets (subsets of size $p+1$) – a much smaller number than $\binom{n}{h}$ in typical applications – and one of these must provide an OLTAD solution for the full data set.

By reversing the order of the two-part search therefore, we can dramatically reduce its computational complexity.

We adapt the selection step to propose the Fast-OLTAD algorithm for linear EIV model, because OLTAD estimator does not need concentration step(C-step).

Another important typical approach to tackle problems is to use heuristics algorithm, such as evolutionary algorithm [11] and Genetic Algorithm (GA) [13].

Genetic Algorithm (GA) [13] is a well established search heuristic in computer science and steadily gaining importance in computational statistics [14].

In fact, many fast algorithms for robust estimators of linear regression model adopt stochastic search tricks [11][21][22] [23]. In light of this ideas, we concentrate on OLTAD estimator's computing and extend genetic algorithm to a framework that is applicable to it.

## 3 The Fast-OLTAD Algorithm for Solving OLTAD Estimator

Based on the above analysis, we proposed a fast method for Solving OLTAD Estimator. Computationally, it is enough to take 500 different initial subsets for the proposed algorithm. Rousseeuw & Van Drissen (1999a) used 500 different initial subsets for computing the minimum covariance determinant estimator.

We describe the algorithm for computing the OLTAD estimator as follows:

*Step 1.* Generate an initial set $H_1 \subset \{1,2,\cdots,n\}$ using the method described as follow: Generate a random ($p$+1)-subset $J$ and compute the exact coefficient $\boldsymbol{\beta}_0$ from $y_i = [\ 1\ \ \boldsymbol{x}_i^T\ ]\boldsymbol{\beta}$, $i \in J$. If the data matrix $\boldsymbol{X}$ is not full rank, then add a random observation into $J$ until it does. Then compute the residuals $r_{0i} = (y_i -$

$[\ 1 \quad \boldsymbol{x}_i^T\ ]\boldsymbol{\beta}_0)/\sqrt{1+\|\boldsymbol{\beta}_0'\|^2}$,$i=1,2,\cdots,n$. wherein $\boldsymbol{\beta}_0'$ is a vector which $\boldsymbol{\beta}_0$ is removed the first element. Set $H_1$ such that$\{\, i \in H_1|\, \left|r_{0(1:n)}\right| \leqslant \left|r_{0(2:n)}\right| \leqslant \cdots \leqslant \left|r_{0(h:n)}\right|\}$.

*Step 2.* Compute the objective function, and repeat Step 1 500 times.

*Step 3.* Set $\boldsymbol{\beta}_0$ corresponding to the minimum objective function as the OLTAD estimator of multiple EIV model.

# 4 The Algorithm for Solving OLTAD Estimator Using Genetic Algorithm

## 4.1 The Core Genetic Algorithm

Genetic algorithms are based on a biological metaphor and the search for the optimal solution is viewed as a competition amongst the population of evolving candidate problem solutions.

The steps of a basic form of the genetic algorithm are given below. These steps are general enough to govern many (perhaps most) modern implementations of genetic algorithm, including those in modern commercial software. Of course, the performance of a GA typically depends greatly on the implementation details, just as with other stochastic optimization algorithms. Some of these practical implementation issues are taken up in the next section.

Core GA Steps:

*Step 0* (Initialization) Randomly generate an initial population of chromosomes and evaluate the fitness function (the conversion of to a function to be maximized for the encoded version of) for each of the chromosomes.

*Step 1* (Parent Selection) Set if elitism strategy is not used; otherwise. Select with replacement parents from the full population (including the elitist elements). The parents are selected according to their fitness, with those chromosomes having a higher fitness value being selected more often.

*Step 2* (Crossover) For each pair of parents identified in Step 1, perform crossover on the parents at a randomly (perhaps uniformly) chosen splice point (or points if using multi-point crossover) with probability. If no crossover takes place (probability), then form two offspring that are exact copies (clones) of the two parents.

*Step 3* (Replacement and Mutation) While retaining the best chromosomes from the previous generation, replace the remaining chromosomes with the current population of offspring from Step 2. For the bit-based implementations, mutate the individual bits with probability ; for real coded implementations, use an alternative form of "small" modification (in either case, one has the option of choosing whether to make the elitist chromosomes candidates for mutation).

*Step 4* (Fitness and End Test) Compute the fitness values for the new population of chromosomes. Terminate the algorithm if the stopping criterion is met or if the budget of fitness function evaluations is exhausted; else return to Step 1.

Genetic algorithms are a general and adaptive framework and ideas can also be found in existing algorithms for statistical applications [14].

## 4.2 Decimal-Integer-Coded Algorithm for OLTAD Estimator

As a first step, we have to appoint the code of the candidate solutions. In order to have a limited number of candidate solutions, we restrict ourselves to candidate solutions uniquely determined by a data subsample of fixed size. In the most common algorithms PROGRESS [15] and Fast-LTS [17] the subsamples are for sound reasons of size $p$. These reasons include the fact, that $p$ linear independent points uniquely define a hyperplane. Additionally, smaller subsamples decrease the possibility of having outliers in the subsample.

We will adopt this in letting for explanatory data $\mathbb{Z}_e = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \cdots, (\mathbf{X}_n, y_n)\} \subset \mathbb{R}^{p+1}$. The set

$$\mathbb{S} = \{\mathbf{s}|\mathbf{s} = (i_1, i_2, \cdots, i_{p+1})\} \tag{13}$$

(where $i_k \in 1, 2, \cdots, n$, $i_q \neq i_r, q \neq r, q, r, k = 1, 2, \cdots, p+1$) is the code of our individuals that is mapped to its phenotype by the function

$$m : \mathbb{S} \to \mathbb{R}^{(p+1)\times(p+1)} \tag{14}$$

$$m(\mathbf{s}) = m((i_1, i_2, \cdots, i_{p+1})) = \begin{bmatrix} \mathbf{X}_{i_1} & y_{i_1} \\ \mathbf{X}_{i_2} & y_{i_2} \\ \cdots & \cdots \\ \mathbf{X}_{i_{p+1}} & y_{i_{p+1}} \end{bmatrix} \tag{15}$$

Thus, we can obtain $\begin{pmatrix} n \\ p+1 \end{pmatrix}$ different possible individuals. The determination of the value of objective function of these individuals comprises two steps:

1. Compute a unique candidate solution hyperplane $H_j$ from the given individual $\mathbf{s}_j \in \mathbb{S}$, where $j = 1, 2, \cdots, popsize, popsize \in \mathbb{N}^+$.

2. Compute one of the objective function $f(\mathbf{s}_j) = \sum\limits_{i=1}^{h_p} \{|r_1|, \cdots, |r_n|\}_{i:n}$ (the residuals $r_i$ are determined by $H_j$ and the given data) depending on the

estimator chosen, $h_p$ with $[(n+p+1)/2 \leqslant h_p \leqslant n$ is a parameter influencing the estimation .

The algorithm that we propose is the following:

Algorithm (Integer-Coded Genetic Algorithm (DICGA) for Trimmed Estimators)

*Step 0* (Initialization) Give the size of population *popsize* and randomly generate an initial population of chromosomes $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{popsize} \in \mathbb{S}$, compute a unique hyperplane $H_j$ from $\mathbf{s}_j$and evaluate the fitness function for each of the chromosomes, let $fitness(\mathbf{s}_j) = \dfrac{1}{1+f(\mathbf{s_j})}, j = 1, 2, \cdots, popsize.$ for the OLTAD estimator.

In general way, the population size $popsize$ can take from 30 to 100, the chromosome $\mathbf{s}_j$ can take $p+1$ numbers from 1 to $n$ randomly.

*Step 1* (Parent Selection) Set if elitism strategy is not used; otherwise. Select with replacement parents from the full population (including the elitist elements). The parents are selected according to their fitness, with those chromosomes having a higher fitness value being selected more often. There are several schemes for the selection process: roulette wheel selection and its extensions: scaling techniques, tournament, elitist models, and ranking methods. A common selection approach assigns a probability of selection $P_j$ to each individual, $j$ based on its fitness value.

$$P_j\{\text{individual } j \text{ is chosen}\} = \frac{fitness(\mathbf{s}_j)}{\sum\limits_{j=1}^{popsize} fitness(\mathbf{s}_j)}, \quad j = 1, 2, \cdots, popsize. \tag{16}$$

Here, we adopt roulette wheel selection.

*Step 2* (Crossover) For each pair of parents identified in Step 1, perform crossover on the parents at a randomly .If no crossover takes place (crossover probability $p_c$ $(0.5 < p_c < 0.95)$), then form two offspring that are exact copies (clones) of the two parents. The crossover operator in details is as follows: Select chromosome $(i_1, i_2, \cdots, i_{p+1})$ and $(j_1, j_2, \cdots, j_{p+1})$, union them and delete redundant same numbers. As a result, we acquire a long chromosome $(k_1, k_2, \cdots, k_m)$ $(p < m \leq 2p+2)$. Subsequently, we attain two new chromosomes taking $(i'_1, i'_2, \cdots, i'_{p+1})$ and $(j'_1, j'_2, \cdots, j'_{p+1})$ from $(k_1, k_2, \cdots, k_m)$ randomly as offsprings.

*Step 3* (Mutation) While retaining the best chromosomes from the previous generation, replace the remaining chromosomes with the current population of offspring from Step 2. For the integer-based implementations, mutate the individual integers with probability $p_m(0.01 < p_m < 0.20)$. If chromosome $(i_1, i_2, \cdots, i_{p+1})$ mutate, randomly select $i_k$ from it and replace $i_k$ by $i'_k$, $i'_k \in \{1, 2, \cdots, n\} \backslash \{i_1, i_2, \cdots, i_{p+1}\}$.

*Step 4* (Fitness and End Test) Compute the fitness values for the new population of chromosomes. Terminate the algorithm if the stopping criterion is met or if the budget of fitness function evaluations is exhausted; else return to Step 1. In general, generation number, running time and no improvement for some iterative times are often as stopping criterion.

The question how to compute a unique hyperplane from a subset of size $p$ remains. As a first step, we compute the hyperplane $H_j$ through the subset of data points. If it does not define a unique hyperplane, we try to add observations in fixed order (e.g. starting with $(\mathbf{X}_1, y_1)$) until it does.

## 5 Simulations And Comparisons

To get an idea of the performance of our new algorithm, we start by applying the decimal-integer-coded genetic algorithm(DICGA) to some small regression data sets taken from [18] and synthetic data.

Table 1 collects the results of our experiments on OLTAD estimator for multiple EIV model. The first column of Table 1 is list the name of each data set, followed by $n$ and $p$, where $n$ is the number of observations and $p$ stand for the number of coefficients including the intercept term. We stayed with the default value of $h = [(n+p+1)/2]$. The next columns show one of best $p$-subset, final best $h$-subset found and running time.

We may conclude that for these small data sets the DICGA gives very accurate results in a short time. In the computer simulation, parameters of DICGA algorithm are set as follows: population size $popsize = 50 + 10 \times p$, crossover rate $p_c = 0.7$, mutation rate $p_m = 0.3$, generation number $N = 30 + 5 \times p$, $p$ is the column of data set $\mathbb{Z}$. We also change parameters to other values, and obtain similar results. The above parameters values can balance the algorithm's accuracy and efficiency in our test situation.

Exact fits to subsets of size $p$ have a special place in the area of high breakdown estimation - these elemental sets have long been used to generate approximations to other high breakdown fit criteria such as OLTS estimator. Unlike OLTS estimator that need C-Step procedure, one of the practical attractions of the OLTAD estimator is the relative ease (compared with the OLTS estimator), because in the case of OLTAD estimator, they yield exact solutions and not just approximations [19].

Since an OLTAD estimator corresponds to an exact fit to some subset of size $p$ [19], the OLTAD estimator is similarly characterized as a two-part problem–identifying the correct subset of size $h$ to cover with the OLTAD estimator, and determining the subset of size $p$ that minimizes the sum of orthogonal absolute deviations to these $h$ cases.

Denote the number of subsets of size $h$ from a sample of size $n$ by $\binom{n}{h}$. There are $\binom{n}{h}$ "elemental" subsets (subsets of size $p$– a much smaller number than $\binom{n}{h}$ in typical applications – and one of these must provide an OLTAD estimator solution for the full data set. For data sets of same size, the number of subsets required for exact evaluation of the OLTAD estimator is far smaller than those required for OLTS estimator.

By reversing the order of the two-part search, therefore, we can dramatically reduce its computational complexity. As a important result, for data sets of same size, the number of subsets required for exact evaluation of the OLTAD estimator is far smaller than those required for OLTS estimator.

Furthermore, in order to explore and test the performance of our new algorithms for OLTAD estimator , we simulate data with $n = 500 + 10 \times p$ data points for $p = 1, \cdots, 30$ from multiple linear EIV model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad (17)$$

where $\beta_0$ is an intercept term and $\varepsilon_i \sim N(0, 1)$ are statistical errors. The parameters $\beta_0, \beta_1, \cdots, \beta_p$ is set to 1, $x_{ij} \sim N(0, 1), j = 1, 2, \cdots, p.$ are the nontrivial explanatory variables. We have introduced outliers in the $x$-direction and $y$-direction by replacing a 25 percent of the $x_{ij}$, $(j = 1, 2, \cdots, p.)$ by that are normally distributed with mean 10 and variance 100.

The DICGA algorithm and Fast-OLTAD algorithm are used to compute OLTAD estimator for above linear EIV model. Based on the computing results, the boxplots of the trimmed OLTAD criterion values v.s the number of regressors are plotted in Figure 1.

In the boxplys of Figure 1, the filled markers represent the results by Fast-OLTAD algorithm and null-box markers represent the results by DICGA algorithm.

From Figure 1, we can conclude that DICGA algorithm achieves better values of the OLTAD-criterion than Fast-OLTAD algorithm in nearly all conducted runs. Of course the DICGA algorithm requires more computation time.

Furthermore, the aspect computing time in the simulations, we find that when the $p$ increase and $n$ is fixed, the computation time mean of DICGA algorithm linearly increases from 1.3 second to 8.2 second, meanwhile, the computation time mean of Fast-OLTAD algorithm linearly increase from 0.05 second to 0.08 second. So, if you want to need higher precision, you can use DICGA algorithm, and if you want save time, you can use Fast-OLTAD algorithm.

Table 1 Performance of DICGA for OLTAD estimator on real data

| data set | $n$ | $p$ | best $p$-subset found | best $h$-subset found | Time($s$) |
|---|---|---|---|---|---|
| Heart | 12 | 3 | 2,6,11 | 1,2,4,5,6, 7,11,12 | 0.7 |
| Phosphor | 18 | 3 | 6,8,18 | 2,4,5,6,8, 9,10,14,15, 17,18 | 0.7 |
| Coleman | 20 | 6 | 2,4,6,10, 12,20 | 1,2,4,5,6, 7,8,9,10,11, 12,17,20 | 0.9 |
| Wood | 20 | 6 | 3,10,12, 15,18,19 | 1,2,3,4,5, 6,10,12,13, 15,18,19,20 | 0.9 |
| Sanlinity | 28 | 4 | 6,15,19, 27 | 2,3,4,6,7, 12,14,15,17, 18,19,20,21, 22, 26,27 | 0.8 |
| Aircraft | 23 | 5 | 11,13,16, 17,19 | 2,4,5,8,9, 10,11,12,13, 16,17,18, 19,21 | 0.8 |
| Delivery | 25 | 3 | 7,10,22 | 2,5,6,7,8, 10,12,13,14, 15,17,21,22, 25 | 0.7 |
| Hawkins | 75 | 4 | 1,5,47,74 | 1,2,3,4,5, 6,7,9,16,17, 18,19,21, 22,24,31, 32,35,36, 37,40,41, 43,45,47, 48,49,50, 51,52,54, 55,59,61, 66, 67,69, 71, 72,74 | 0.8 |
| Bushfire | 38 | 5 | 9,12,23, 28,37 | 7,8,9,10, 12,13,15, 17,19,20, 21 | 0.9 |

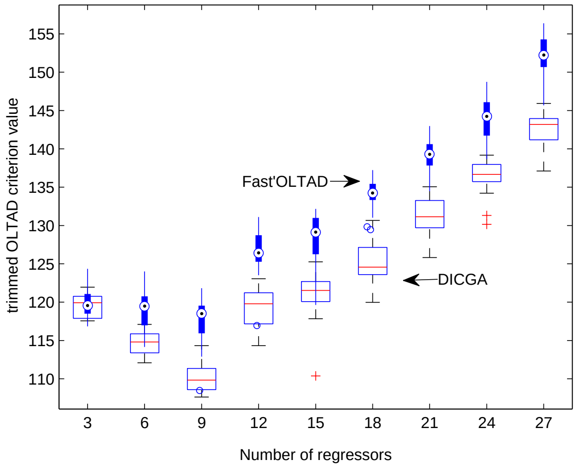

Figure 1: The boxplot of trimmed OLTAD criterion values vs. number of regressors computed by DICGA and Fast-OLTAD algorithms

All computational experiments where performed on a Intel Core Duo 2.4GHz computer with 2GB of memory running MATLAB 2009a and Windows XP Sp2.

## 6 Conclusion

High breakdown estimators in big data sets are a challenging problem. Many high breakdown estimators are in all likelihood not computable exactly in high dimensional regressor spaces. The common heuristics to compute solutions in these cases work with subsample versions of the estimators. By comparison with $0-1$ integer-coded genetic algorithm [14] and other evolution algorithms [11][23], the individuals of decimal-integer-coded genetic algorithm is shorter, especially when sample data number $n$ is big, so DICGA occupies less computer's memory and it is less time-consuming.

Comparing with OTLS estimator of multiple linear EIV model, our new robust OLTAD estimator is easier to implementation in computer. What more, our new robust OLTAD estimator is less time-consuming, because it needed the C-Step procedures. Since the C-Step procedures in computing OTLS estimator needs many iterations and redetermining the multiple linear EIV model in each iteration, so computing OTLS estimator is more complicated and need more computing time.

In this paper, a new and simple robust estimator and corresponding algorithms for multiple linear EIV model are proposed. The research enriches the theory and method study of multiple linear EIV model and at the same time, it provides a new choice for engineers and scientific researchers who use linear EIV model in their work.

**Acknowledgements:** The research was supported by the the Natural Science Foundation of Hebei Province of China( grant No. A2011408006). The authors are thankful to anonymous referees for their constructive suggestions and critical comments, which led to this improved version.

*References:*

[1] M. L. Brown, Robust line estimation with errors in both variables, *J. Am. Stat. Assoc.*, Vol. 77, 1982, pp.71–79 .

[2] R. H. Zamar, Robust estimation in the errors-in-variables model, *Biometrika*,Vol. 76, No.1,1989, pp. 149–160 . h

[3] R. H. Zamar, Bias robust estimation in the orthogonal regression, *Ann. Appl. Stat.*, Vol. 20, No. 4, 1992, pp. 1875–1888.

[4] C. Croux and G. Haesbroeck, Principal component analysis based on robust estimators of the covariance or correlation matrix:Influence functions and effciencies, *Biometrika*, Vol. 87, No. 3, 2000, pp.603–618 .

[5] M. Fekri and A. Ruiz-Gazen, Robust weighted orthogonal regression in the errors-in-variables model, *J. Multivariate. Anal.*, Vol.88, No. 1, 2004, pp. 89–108 .

[6] K. M. Jung, Least trimmed squares estimator in the Errors-in-Variables model, *J. Appl. Stat.*, Vol. 34, No. 3, 2007, pp. 331–338.

[7] C. Croux , M. Fekri and A. Ruiz-Gazen. Fast and robust estimation of the multivariate errors in variables model. *Test*, Vol.19, No.2, 2009, pp. 286–303.

[8] T. Hu and H. J. Cui, t-type estimators for a class of linear errors-in-variables models, *Stat. Sinic.*, Vol. 19, No. 3, 2009, pp.1013–1036.

[9] Y. Dodge, *Statistical data analysis based on the L1-norm and related methods*, Boston, Basel, Berlin: Birkhäuser, 2002, PP.109–123.

[10] I. Wegener, *Complexity Theory*, Springer, Berlin Heidelberg,2005.

[11] R. Nunkesser and O. Morell, An evolutionary algorithm for robust regression, *Comput. Stat. Data An.*, Vol.54, No.12, 2010, pp. 3242-3248.

[12] G. Willemsa, S. Van Aelst, Fast and robust bootstrap for LTS, *Comput. Stat. Data An.*, Vol. 48,No.4, 2005, pp. 703–715.

[13] S. Forrest, Genetic algorithms: Principles of natural selection applied to computation, *Science*,Vol.261, 1993, pp.872–878.

[14] S. Chatterjeea, M. Laudatoa and L. A. Lynchb , Genetic algorithms and their statistical applications: an introduction, *Comput. Stat. Data An.*, Vol. 22, No.6, 1996, pp.633–651.

[15] P. J. Rousseeuw and M. Hubert, Recent developments in progress. In: Dodge, Y. (Ed.) ,$L_1$-*Statistical Procedures* $L_1$-*Statistical Procedures and Related Topics.* Vol. 31 of Lecture Notes-Monograph Series. Institute of Mathematical Statistics,1997.

[16] D. M. Hawkins, The feasible solution algorithm for least trimmed squares regression,*Comput. Stat. Data An.*, Vol.17, No.2, 1994, pp. 185-196.

[17] P. J. Rousseeuw and Van K. Driessen, Computing lts regression for large data sets, *Data. Min. Knowl. Disc.*, Vol.12, No.1, 2006, pp.29–45.

[18] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outliers Detection*, John Wiley and Son ,1987.

[19] S. P. Feng, Optimal solution on weighted total least absolute deviation(In Chinese),*Journal of University of Science and Technology of China*, Vol.39, No.12, 2009, pp.1260–1264.

[20] H. J. Cui, Asymptotic properties of Generalized LAD estimators in EV model(In Chinese), *Science in China (Series A)*,Vol. 27, No.2, 1997, pp.119–131.

[21] G. Willems and S. Van Aelst, Fast and robust bootstrap for LTS, *Comput. Stat. Data An.*, Vol. 48, No.4, 2005, pp. 703–715.

[22] F. Torti, D. Perrotta, A. C. Atkinson and M. Rianid, Benchmark testing of algorithms for very robust regression: FS, LMS and LTS, *Comput. Stat. Data An.*, Vol.56, No 8, 2012, pp.2501–2512.

[23] M. Hofmann, C. Gatu and E. J. Kontoghiorghes, Efficient algorithms for computing the best subset regression models for large-scale problems, *Comput. Stat. Data An.*, Vol.52, No.1, 2007, pp.16-29.