UNIVERZITA KARLOVA V PRAZE

FAKULTA SOCIÁLNÍCH VĚD

Institut ekonomických studií

Bakalářská práce

2011 **Věra Bludská**

UNIVERZITA KARLOVA V PRAZE

FAKULTA SOCIÁLNÍCH VĚD

Institut ekonomických studií

Věra Bludská

White Test for the Least Weighted Squares

Bakalářská práce

Praha 2011

Autor práce: **Věra Bludská**

Vedoucí práce: **Prof. RNDr. Jan Ámos Víšek CSc.**

Rok obhajoby: 2011

Bibliografický záznam

BLUDSKÁ, Věra. *White Test for the Least Weighted Squares.* Praha, 2011. 39 s. Bakalářská práce (Bc.) Univerzita Karlova, Fakulta sociálních věd, Institut ekonomických studií. Vedoucí bakalářské práce Prof. RNDr. Jan Ámos Víšek CSc.

Abstract

The Least Weighted Squares (LWS) is a robust method for computing coefficients in linear regression models. An inherent problem of LWS is the complexity of its estimator and, consequently, the lack of an analytical solution or fast exact algorithms for its evaluation. To remedy this situation a novel exact algorithm running in polynomial time has been proposed. The algorithm implemented in MATLAB programming language has been employed for testing computationally more efficient non-exact LWS methods. In addition to many potential uses of LWS in robust econometrics (*e.g.* outlier diagnostics) the method has been applied to the problem of regression estimation in the presence of heteroscedasticity. It has been demonstrated that the combined use of the LWS estimator and White's test for heteroscedasticity significantly improves the efficiency of the robust regression estimation.

Keywords

least weighted squares, algorithm, CP-LWS, heteroscedasticity

Abstrakt

Nejmenší vážené čtverce (LWS) jsou robustní metodou pro získání koeficientů lineárních regresních modelů. Hlavní nevýhoda této metody (oproti klasické metodě nejmenších čtverců) spočívá ve složitosti příslušných rovnic, jejichž analytické řešení neznáme, a také v neznalosti dostatečně rychlých algoritmů pro numericky přesné výpočty LWS. Jedním z hlavních výsledků této práce je navržení nového deterministického LWS algoritmu běžícího v polynomiálním čase. Tento algoritmus, implementovaný v programovém prostředí MATLAB, poskytuje přesné (numericky exaktní) hodnoty LWS odhadů a byl použit pro testování rychlých aproximativních metod. Kromě standardního využití LWS v robustní ekonometrii (např. detekce odlehlých pozorování) byla tato metoda použita pro odhalení heteroskedasticity disturbancí při robustní regresní analýze dat. V práci je názorně ukázáno, že spojení LWS s Whiteovým testem signifikatně vylepšuje vydatnost robustních odhadů.

Klíčová slova

nejmenší vážené čtverce, algoritmus, CP-LWS, heteroskedasticita

Prohlášení

Prohlašuji, že jsem předkládanou práci zpracovala samostatně a použila jen uvedené prameny a literaturu. Prohlašuji, že práce nebyla využita k získání jiného titulu.

V Praze dne 20. května 2011 Věra Bludská ....

Poděkování

Na tomto místě bych ráda poděkovala Prof. RNDr. J.Á.Víškovi CSc. za cenné připomínky a trpělivost.

Contents

1. Introduction

Robust regression analysis deals with gross errors and deviations from model assumptions that are potentially dangerous to many "classical" statistical procedures such as the ordinary least squares (OLS). Gross errors (misplaced decimal points, data transmission errors, etc.) usually lead to the occurrence of outliers. There are two approaches to the problem of outlying observations: new statistical methods that are not so sensitive to outliers, and outlier diagnostics (*e.g.* by a visual inspection of data), both are the robust statistics methods in broader sense. Several strategies have been devised including semiparametric and nonparametric methods that relax assumptions made in standard statistical approaches. These methods, however, tend inevitably to be less efficient and require larger amount of data. Another strategy is to make estimators more robust to small deviations from parametric assumptions. As a result, the robust parametric approaches are less prone to the contamination of data. The Least Trimmed Squares (LTS) was among the first robust estimators with a high breakdown point [Rousseeuw 1987]. LTS keeps many advantages of the least squares and possesses a tunable breakdown point (resistance to outliers) up to 50%. It should be stressed, however, that requirement of highest possible breakdown point often contradicts some other basic requirements of robust estimator, efficiency in particular. The LTS estimator has also a very high subsample sensitivity, *i.e.* removal of a single observation can profoundly change the result. The Least Weighted Squares (LWS) as a generalization of both OLS and LTS (these are limiting cases of LWS) has all desirable properties of LTS, but offers also room for improvement – *e.g.* above mentioned subsample sensitivity.

An inherent problem of LWS (and many other robust methods) is the complexity of its estimator, and therefore the lack of an analytical solution or fast exact algorithms for its evaluation. Hence, evaluations of robust estimates rely on approximations with varying reliability and this fact, of course, limits the applicability of robust methods and their penetration into standard statistical packages. As a consequence, a typical user has little experience with behavior of robust methods. Another problem stems from the approximative nature of robust estimates. In numerical experiments, for example, it might be difficult in some cases to distinguish behavior of the LWS estimator from behavior of its approximation. Thus, one of main goals of this work is to contribute to LWS-related research by developing either a sufficiently fast exact algorithm or algorithms for finding approximate solutions with a high (controllable) accuracy.

The second subject of this work is related to the presence of heteroscedastic disturbances in the linear regression model. This situation arises quite often in econometric studies: data aggregating over some regions, explanatatory variables measured with random errors, models with randomly varying coefficients, autoregressive models, employing probit, logit or counting models, etc. (for comprehensive list see [Víšek 2004]). The presence of heteroscedastic disturbances strongly influences the efficiency of classical statistical procedures, and therefore a more resistant estimator against heteroscedasticity should be employed in such situations. The LWS estimator is certainly a plausible candidate to cope with heteroscedastic disturbances.

This work is organized as follows: the remaining part of Section 1 introduces the regression model and data modeling used throughout the text. For reasons mentioned in Section 1.1 we restrict ourselves to the simplest linear regression with only one explanatory variable and intercept, data modeling varying only the most important model parameters such as dataset size, variances of disturbances, number of outliers and leverage points, contamination by a second regression model. These parameters are essential for performance testing of implemented robust regression algorithms. Section 2 contains the description of implemented LWS methods. Some of them were taken from literature and coded in MATLAB programming language (P-LWS, FastLWS). Two newly proposed algorithms were developed by the author of this work and both are based on the concept of geometrically feasible permutations which is the result presented here for the first time. To author's knowledge, the proposed algorithm for finding the exact solution of the Least Weighted Squares (LWS) is the first exact algorithm running in polynomial time. The second (non-exact) variant of the same algorithm has comparable computational efficiency to the fastest available codes but much higher reliability (Fast CP-LWS). Finally, the algorithm based on a direct stochastic optimization by Differential Evolution (DE) is, as far as the author knows, the first implementation of DE in context of LWS. Section 3 is devoted to testing and comparison of methods described in the previous section. Section 4 is focused on solving one of the most challenging problem in robust econometrics – robust estimation in the presence of heteroscedasticity. The last section summarizes the main results of this work and discusses an outlook of LWS solvers in econometric applications.

1.1 Regression Model

Let us define the linear regression model

$$Y_i = x_i^T \beta^0 + e_i \qquad \text{(i = 1, 2, …, } n\text{)}, \tag{1.1}$$

with n observations on the response variable Y_i and the deterministic vector of explanatory variables $x_i = (x_{i1},...,x_{ip})^T \in \mathbb{R}^p$, where p is a number of regression coefficients, $\mathbb{R}^p$ is the p-dimensional Euclidean space, $\beta^0 = (\beta_1^0,...,\beta_p^0)^T \in \mathbb{R}^p$ is a vector of the "true" regression coefficients and e_i is a stochastic disturbance (error term). In a model with intercept the first (additional) component in all vectors x_i (i = 1, 2, ...,n) is set to 1 and the vector of regression coefficients includes the intercept, β_0^0, *i.e.* $\beta^0 = (\beta_0^0,...,\beta_{p-1}^0)^T \in \mathbb{R}^p$. Residuals are given as

$$r_i(\hat{\beta}) = Y_i - x_i^T \hat{\beta} \ , \tag{1.2}$$

where $\hat{\beta}$ is the vector of regression coefficient estimates. Thus, the residuals $r_i(\hat{\beta})$ represent the difference between the observed value Y_i and the estimated value ($\hat{Y}_i = x_i^T \hat{\beta}$).

The model (1.1) can be written also in a matrix form as

$$Y = X\beta^0 + e, \tag{1.3}$$

where Y is a vector of type ($n \times 1$), X is a design matrix of size ($n \times p$) with rows consisting of vectors x_i^T

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{1.4}$$

and e is a vector of disturbances of type ($n \times 1$).

Throughout the text we will consider a model with only two regression parameters including intercept (β_0, β_1), because graphical presentation of results in parameter space is much simpler to visualize. Generalization of our results to the p-dimensional space is relatively straightforward and will be briefly discussed in Section 5. Nevertheless, for p = 2 it is easier to explain the methods in question and also computations will be less demanding.

The most common way how to estimate unknown parameters in the regression model (1.1) is the ordinary least squares (OLS) method. This method minimizes the sum of squared residuals, so we obtain regression coefficient estimates as

$$\hat{\beta}^{(OLS,n)} = \arg\min \sum_{i=1}^{n} (Y_i - x_i^T \beta)^2 \; . \tag{1.5}$$

The definition (1.5) gives us the estimate $\hat{\beta}^{(OLS,n)}$ in an explicit form

$$\hat{\beta}^{(OLS,n)} = (X^T X)^{-1} X^T Y \; . \tag{1.6}$$

To guarantee the plausible properties of $\hat{\beta}^{(OLS,n)}$ we usually adopt the following assumptions:

- the mean value of disturbances is equal to zero ($E(e_i) = 0$, i = 1,2,…,n) which is equivalent to $E(Y_i) = x_i^T \beta^0$, yielding unbiasedness (see Lemma 1 below),
- $\mathrm{var}(e_i) = E[(e_i - E(e_i))^2] = E(e_i^2) = \sigma^2$, i = 1, 2, …, n, $0 < \sigma < \infty$ (all e_i have the same variance which we call homoscedasticity),
  if $\mathrm{var}(e_i) = \sigma^2$ then also $\mathrm{var}(Y_i) = \sigma^2$,
- the covariance between any pair of random errors is equal to zero ($\mathrm{cov}(e_i, e_j) = 0$ for $i \neq j$, i,j = 1, 2, …, n) which together with homoscedasticity implies that $\hat{\beta}^{(OLS,n)}$ is the best linear estimator,
- matrix X is deterministic and has full rank (rank(X) = p and thus $(X^T X)$ is regular).

Let us recall some important properties of the estimator $\hat{\beta}^{(OLS,n)}$.

Lemma 1: *Let* $\{e_i\}_{i=1}^{n}$ *be a sequence of random variables, such that* $E(e_i) = 0$ *and* $E(e_i e_j) = \delta_{ij}\sigma^2$, *where* $\delta_{ij} = 0$ *for* $i \neq j$ *and* $\delta_{ij} = 1$ *for* $i = j$, $0 < \sigma < \infty$. *Then* $\hat{\beta}$ *is the best linear unbiased estimator. If moreover* $(X^T X) = O(n), (X^T X)^{-1} = O(n^{-1})$ *and* e *are independent, then* $\hat{\beta}$ *is consistent. If further* $\lim_{n\to\infty} \frac{1}{n}(X^T X) = Q$ *is regular, then* $\mathcal{L}(\sqrt{n}(\hat{\beta} - \beta^0)) \xrightarrow[n\to\infty]{} N(0, \Sigma)$ *where* $\Sigma = \mathrm{cov}(\sqrt{n}(\hat{\beta} - \beta^0)) = \sigma^2 Q^{-1}$ *(asymptotic normality).*

Lemma 2: *Let* $\{e_i\}_{i=1}^{n}$ *be a sequence of independent identically distributed (iid) random variables, such that* $\mathcal{L}(e_i) = N(0, \sigma^2)$, $0 < \sigma < \infty$. *Then* $\hat{\beta}$ *is the best unbiased estimator.*

For details see [Víšek 1997].

OLS method is based on minimization of the sum of squared residuals which means high sensitivity to leverage points and outliers (points that are distant from the rest of the sample set). It can lead to false OLS estimation, far away from the “true” β^0. That is why we are presenting here other estimators based on weighted least squares concept.

1.2 Data Modeling

The purpose of this section is to establish a sufficiently flexible model for simulating a various degree of data contamination. The model will be used for analyzing properties of employed robust estimators such as breakdown point, loss of efficiency, etc. We consider two kinds of data contamination: (i) contamination by outliers and leverage points, i.e. by observations that deviate markedly from the underlying regression model (1.1) and (ii) by observations that come from another (different) regression model. A typical example of both kinds of data contamination is depicted in Figure 1.

The data in our model are generated by the following submodels:

- model described by (1.1) with

  $\beta \sim N(0,1)$, $\quad x_i^T \sim (0,1)$, $\quad e_i \sim N(0,\sigma_i^2)$.

  The σ_i values are the only parameters of the model. The elements of the vector β are drawn from the standard normal distribution, each element of the vector x_i^T is randomly chosen from the interval (0,1), and disturbances are drawn from normal distribution $N(0,\sigma_i^2)$, *i.e.* the disturbances are heteroscedastic in general (if all σ_i are set to the same value, disturbances are homoscedastic).

- model with outliers

  $Y_i = x_i^T\beta^0 + \Delta_o + e_i$, where

  $\beta \sim N(0,1)$, $\quad x_i^T \sim (0,1)$, $\quad e_i \sim N(0,k_o\,\sigma_i^2)$.

  The σ_i values are taken from the model described above, the constants Δ_o and $k_o > 0$ are parameters of the model. The Δ_o parameter represents a constant shift of Y_i values with respect to (1.1) and k_o scales the variance of disturbances.

- model with leverage points

  $\beta \sim N(0,1)$, $\quad x_i^T = \Delta_l + k_l\,\tilde{x}_i^T$, where $\tilde{x}_i^T \sim (0,1)$, $\quad e_i \sim N(0,\sigma_i^2)$.

  The σ_i values are taken the same as in previous models, the constants Δ_l and k_l are parameters of the model. The Δ_l parameter represents a constant shift of x_i^T values and $|k_l| > 1$.

- model with outlying leverage points

This model combines two previous models, the model with outliers and the model with leverage points, by setting the Δ_o, k_o, Δ_l, and k_l parameters simultaneously.

- another model (type (ii) contamination) with

  $\beta_2 \sim N(0,1)$, $\quad x_i^T \sim (0,1)$, $\quad e_i \sim N(0,\sigma_2^2)$.

Throughout this work the parameters $\Delta_o, k_o, \Delta_l,$ and k_l were set to 5, 100, 2, and 5, respectively. For a given n our model is defined by a number of points generated by each submodel: number of outliers (n_o), leverage points (n_l), outlying leverage points (n_{ol}), and points drawn from the second regression model (n_c).

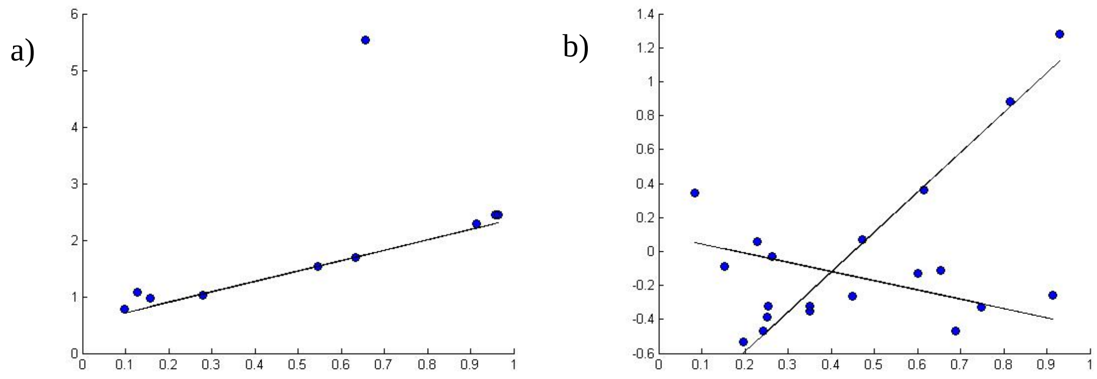

Figure 1
Example of the type (i) and type (ii) of data contamination. a) Regression model with outliers, (ii) contamination of data from the different regression model.

2. The Least Weighted Squares

LWS is an alternative method to OLS for computing regression coefficient estimates in a linear regression model. This method was introduced by [Víšek 2002a,b] (and definition is recalled below). It is a robust method where individual observations have different weights in the model. The presence of outliers in the model does not necessarily mean false estimation if weights are chosen in an appropriate way. The least weighted squares (LWS) estimator fulfills all requirements of modern paradigm of point estimation [Víšek 2002a,b]. Due to mathematical complexity of the underlying theory only main properties of the LWS estimator ($\hat{\beta}^{LWS}$) will be summarized here [Víšek 2000]:

- $\sqrt{n}$ -consistent and asymptotically normal,
- reasonably high efficient,
- scale and regression-equivariant,
- it has a high (controllable) breakdown point.

Definition 1 An estimator T of β is called scale-equivariant if satisfies

$$T\,[(x_i^T, \alpha\; y_i)] = \alpha\; T\,[(x_i^T, y_i)]\,. \tag{2.1}$$

Definition 2 An estimator T of β is called regression-equivariant if for any vector $\beta \in R^p$ satisfies

$$T\,[(x_i^T, y_i + x_i^T \beta)] = T\,[(x_i^T, y_i)] + \beta \tag{2.2}$$

The breakdown point tells us, how much data could cause an estimation failure. It is a measurement of robustness. Robust methods with high breakdown point are able to eliminate the influence of outliers and leverage points.

In the next subsection we will define the LWS estimator using notation introduced in [Víšek 2002a].

2.1 Definition of the LWS estimator

The LWS estimator, $\hat{\beta}^{LWS}$, is given by

$$\hat{\beta}^{(LWS,n,\psi)} = \underset{\beta \in R^p}{arg\ min} \quad \sum_{l=1}^{n} \psi\left(\frac{l-1}{n}\right) r_{(l)}^2(\beta), \tag{2.3}$$

where ψ is a continuous non-increasing function $\psi : [0,1] \to [0,1]$ such that $\psi(0)=1$ and $\psi(1)=0$ (for more precise definition see [Víšek 2002a]) and the order statistics of squared residuals, $r_{(l)}^2(\beta)$, satisfy

$$0 \le r_{(1)}^2(\beta) \le r_{(2)}^2(\beta) \le \ldots \le r_{(n)}^2(\beta). \tag{2.4}$$

The definition (2.3) is similar (albeit conceptually very different) to the weighted least squares (WLS) estimator

$$\hat{\beta}^{(WLS,n,w)} = \underset{\beta \in R^p}{arg\ min} \quad \sum_{l=1}^{n} w_l\, r_l^2(\beta). \tag{2.5}$$

The WLS estimator can be expressed in a compact matrix form using the explicit formula

$$\hat{\beta}^{WLS} = \left(X^T W X\right)^{-1} X^T W Y, \tag{2.6}$$

where *W* is a diagonal square matrix of size *n* with weights (w_i) on the diagonal.

Now let us define permutation of integers 1, 2, 3,..., n

$$g = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ g_1 & g_2 & g_3 & \cdots & g_n \end{pmatrix}$$

so that $0 \le r_{g_1}^2(\beta) \le r_{g_2}^2(\beta) \le \ldots \le r_{g_n}^2(\beta)$. There always exists inverse permutation $h = g^{-1}$ satisfying $g \cdot h = h \cdot g = e$, where $e = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ 1 & 2 & 3 & \cdots & n \end{pmatrix}$ is the identity permutation. The definition (2.3) can be rewritten as

$$\hat{\beta}^{(LWS,n,\psi)} = \underset{\beta \in R^p}{arg\ min} \quad \sum_{l=1}^{n} \psi\left(\frac{l-1}{n}\right) r_{g_l}^2(\beta) = \underset{\beta \in R^p}{arg\ min} \quad \sum_{l=1}^{n} \psi\left(\frac{h_l-1}{n}\right) r_l^2(\beta). \tag{2.7}$$

So if we put $\tilde{w}_l = \psi\left(\frac{h_l-1}{n}\right)$, we obtain

$$\hat{\beta}^{(LWS,n,\psi)} = \underset{\beta \in R^p}{arg\ min} \quad \sum_{l=1}^{n} \tilde{w}_l\, r_l^2(\beta), \tag{2.8}$$

which is formally equivalent to (2.5) and consequently

$$\hat{\beta}^{LWS} = \left(X^T \tilde{W} X\right)^{-1} X^T \tilde{W} Y, \tag{2.9}$$

where $\tilde{W}$ is a diagonal square matrix of size n with $\tilde{w}_i$ on the diagonal.

The difference between the LWS and WLS estimators (cf. (2.6) and (2.9)) stems from the fact that LWS finds the solution of (2.7) over all possible orderings of weights (for all $h \in \mathbb{H}^n$). Thus, (2.9) does not represent an explicit formula for the LWS estimator.

2.2 Computational Algorithms

A key part of establishing any estimator is development of fast and reliable algorithms and their implementations. Thus, large effort has been devoted to further improvement of the current LWS implementations. Sections 2.2.1 and 2.2.2 deal with previously proposed LWS - permutation, branch&bound, and fast iterative inexact algorithms described in [Skuhrovec 2009]. In Section 2.2.3 we propose a novel exact approach denoted as constrained permutation LWS (CP-LWS) algorithm. CP-LWS represents a vast improvement over the permutation and branch&bound methods (see Figure 9). The main problem of currently implemented exact algorithms stems from exceedingly high CPU time demands rendering them impractical for larger (>15) datasets. The computational requirements of both methods grow as n! (factorial time complexity, cf. solving the traveling salesman problem via brute-force search). CP-LWS is of polynomial time, or more precisely, it has an average-case polynomial time complexity.

2.2.1 Permutation LWS (P-LWS) Algorithm

The P-LWS method solves the optimization problem (2.8) by the brute-force search technique. The algorithm minimizes the corresponding objective function

$$S^2(\beta, h) = \sum_{l=1}^{n} \psi\left(\frac{h_l - 1}{n}\right) r_l^2(\beta) \tag{2.10}$$

for all $h \in \mathbb{H}^n$. From $\left|\mathbb{H}^n\right| = n!$ (size of $\mathbb{H}^n$) follows that the time complexity of the P-LWS algorithm is $O(n!)$. The algorithms with exponential or factorial time complexity are usually highly impractical and can be only used for very small datasets (n < 15).

The P-LWS algorithm is given by the following scheme:

(1) - For each $h \in \mathbb{H}^n$ do steps (2)-(5) :

(2) - Calculate $\tilde{w}_l = \psi\left(\frac{h_l - 1}{n}\right)$ and set up the $\tilde{W}$ matrix.

(3) - Calculate $\beta = \left(X^T \tilde{W} X\right)^{-1} X^T \tilde{W} Y$.

(4) - Evaluate the objective function $S^2(\beta, h)$.

(5) - Is $S^2(\beta, h)$ minimum? If yes: $S^2_{LWS} = S^2(\beta, h)$, $\hat{\beta}^{LWS} = \beta$.

The branch&bound algorithm (BB-LWS) is roughly 100 times faster than P-LWS (for description of the BB-LWS see [Skuhrovec 2009]). Considering $O(n!)$ complexity of P-LWS, the speed up of BB-LWS is hardly any improvement for larger datasets (n > 15).

2.2.2 FastLWS Algorithm

The FastLWS algorithm (see *e.g.* [Skuhrovec 2009]) is an inexact method for solving (2.8). The method is based on random sampling of parameter space generated by the ordinary least squares (OLS) estimator for selected subset of p observations (the original idea comes from the Rousseeuw's FastLTS algorithm [Rousseeuw 2002]). The FastLWS algorithm is summarized by the following scheme:

(1) - Randomly select p rows of the design matrix, $\tilde{X}$ is $(p \times p)$ submatrix of X (assumed to be regular matrix), $\tilde{Y}$ is $(p \times 1)$ submatrix of Y.

(2) - Calculate $\hat{\beta}^{OLS} = \left(\tilde{X}^T \tilde{X}\right)^{-1} \tilde{X}^T \tilde{Y}$ and corresponding residuals $r = Y - X^T \hat{\beta}^{OLS}$.

(3) – Calculate g satisfying $0 \le r^2_{g_1}(\beta) \le r^2_{g_2}(\beta) \le \ldots \le r^2_{g_n}(\beta)$ and weights $\tilde{w}_l = \psi\left(\frac{h_l - 1}{n}\right)$ ($h = g^{-1}$) and set up the $\tilde{W}$ matrix.

(4) - Calculate $\beta = \left(X^T \tilde{W} X\right)^{-1} X^T \tilde{W} Y$ and residuals $r = Y - X^T \beta$.

(5) - Is $0 \le r^2_{g_1}(\beta) \le r^2_{g_2}(\beta) \le \ldots \le r^2_{g_n}(\beta)$ satisfied?

If yes : $\hat{\beta}^{LWS} = \beta$ (the best estimate, not necessarily global minimum),

otherwise go to (3)

(6) - $N_{iter} = N_{iter} + 1$ (number of FastLWS iterations/evaluations)

If $N_{iter} <$ max iterations, go to (1).

2.2.3 Constrained Permutation LWS (CP-LWS) Algorithm

The proposed CP-LWS algorithm belongs to the class of exact LWS methods. The idea of CP-LWS is based on an observation that many permutations $h \in \mathbb{H}^n$ (or equivalently the order statistics of squared residuals, see Section 2.1 for more details) cannot be geometrically realized for a given set of points ($X_i = [x_i, y_i],\ i = 1, \ldots, n$).

Definition 3 (Set of geometrically feasible permutations). For each point in a parameter space $\beta = (\beta_0, ..., \beta_{p-1})^T \in R^p$ there is a permutation $g = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ g_1 & g_2 & g_3 & \cdots & g_n \end{pmatrix}$ satisfying $0 \le r^2_{g_1}(\beta) \le r^2_{g_2}(\beta) \le \ldots \le r^2_{g_n}(\beta)$, where $r_{g_i}(\beta)$ are residuals given by $r_{g_i}(\beta) = Y_{g_i} - x^T_{g_i}\beta$. The set of all permutations $h = g^{-1}$ will be denoted as $\mathbb{H}^n_f$.

Since CP-LWS algorithm explores only geometrically feasible permutations $h \in \mathbb{H}^n_f$, where $\mathbb{H}^n_f \subset \mathbb{H}^n$, the time complexity of CP-LWS depends on the size of $\mathbb{H}^n_f$ instead of $\mathbb{H}^n$. Hence, the main task in CP-LWS is a computationally efficient construction of $\mathbb{H}^n_f$. This task can be, at least in principle, carried out by sampling of the parameter space (β-space), which is actually the basis of the FastLWS algorithm described in Section 2.2.2. To author's knowledge there is no sampling algorithm that would guarantee to obtain all $h \in \mathbb{H}^n_f$ in a finite number of steps and therefore any method based on the sampling of the β-space does not belong to the class of exact LWS algorithms. In the following we will approach this problem from a different perspective. Let us consider a set of points in β-space, $\mathbb{M}_g \subset \mathbb{R}^p$, satisfying the condition $0 \le r^2_{g_1}(\beta) \le r^2_{g_2}(\beta) \le \ldots \le r^2_{g_n}(\beta)$ for a given permutation g ($g^{-1} \in \mathbb{H}^n_f$). Obviously, for any $\beta \in \mathbb{M}^o_g$ ($\mathbb{M}^o_g$ denotes an interior of $\mathbb{M}_g$) the following inequalities hold

$$0 \le r^2_{g_1}(\beta) < r^2_{g_2}(\beta) < \ldots < r^2_{g_n}(\beta).$$

At the boundary of $\mathbb{M}_g$, $\beta \in \mathbb{M}_g \cap \mathbb{M}_{g'}$, at least two residuals must equal:

$$\exists i, j : r^2_{g_i}(\beta) = r^2_{g_j}(\beta),\ r^2_{g'_i}(\beta) = r^2_{g'_j}(\beta),\ i \ne j,$$

where permutation g' differs from g only by i-th and j-th elements ($g_i = g'_j$ and $g_j = g'_i$). Therefore, the β-space can be decomposed into a finite number of $\mathbb{M}$ sets corresponding to all geometrically feasible permutations. Along these lines we will elaborate this approach further. For the sake of simplicity, however, we will consider the $p = 2$ case only.

As mentioned above we will focus on the boundary conditions $r_i^2(\beta) = r_j^2(\beta)$ determining the number of feasible permutations. Let us select two arbitrary X_i and X_j points and construct their midpoint. Any line crossing the midpoint has the same residuals in absolute value with respect to these two points, *i.e.* the corresponding β satisfies the condition $r_i^2(\beta) = r_j^2(\beta)$ (see Figure 2). Moreover, it can be easily shown that the set of

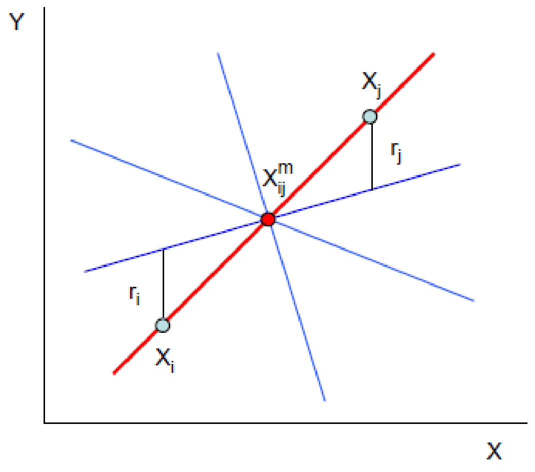

Figure 2
Construction of lines with $r_i = -r_j$ (in blue) for arbitrary selected points.

all lines in XY space crossing the midpoint corresponds to a single line in β-space. Similarly, each line that is parallel with the line X_iX_j has the same two residuals (green lines in Figure 3a). Notice, that there is no other line in XY space satisfying the condition $r_i^2 = r_j^2$.

Now we can show that the set of the lines crossing the midpoint (blue lines in Figure 3a) generate a single line in β-space, let us designate it as L_{ij}^m (blue line in Figure 3b). Note that each line in XY space corresponds to one point in β-space. We can describe this line as $\beta_0 = -\beta_1 x_{ij}^m + y_{ij}^m$, where $[x_{ij}^m, y_{ij}^m] = X_{ij}^m$. Similarly, the set of the parallel lines (green lines in Figure 3a) generate a single line in β-space, we designate it as L_{ij}^s (green line in Figure 3b).

Now it should be realized that these two lines (L_{ij}^m, L_{ij}^s) crosses each other creating four separate areas ($\mathbb{M}$ sets) in β-space (Figure 3b). Each point on the L_{ij}^m and L_{ij}^s line holds the condition $r_i^2 = r_j^2$ and their crossing point (marked red in the Figure 3b) has residuals equal to zero, because it has to comply with $r_i = r_j$ and $r_i = -r_j$ simultaneously (this point

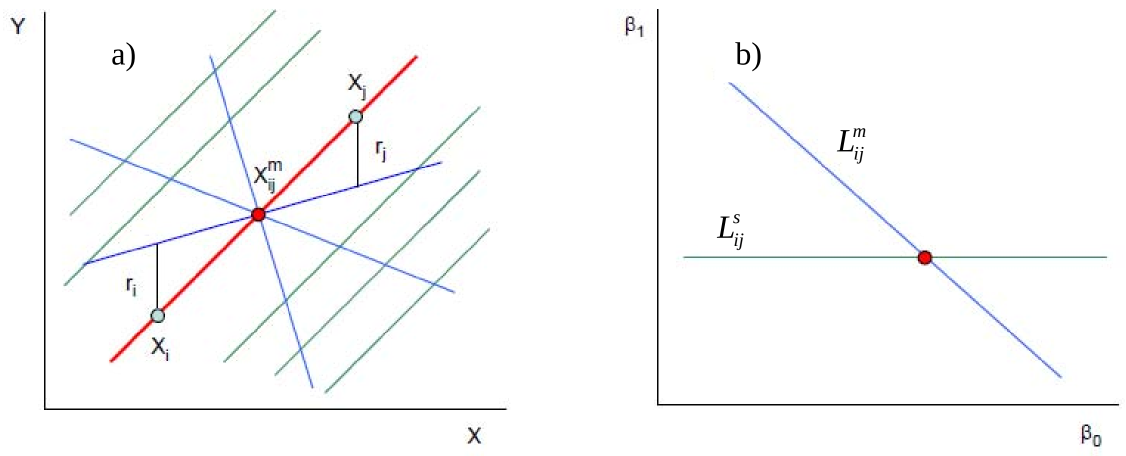

Figure 3
a) Construction of lines with $r_i = r_j$ (green) and $r_i = -r_j$ (blue) in XY space and
b) their representation in β-space. The intersection point in β-space (marked red) corresponds to the line in XY space with $r_i = 0, r_j = 0$ (red).

represents the X_iX_j line in XY space). If we choose any area generated by L_{ij}^m and L_{ij}^s lines and cross one of these lines, in the second area there must be different sequence of squared residuals. So if all points in the interior of original area comply with $r_i^2 < r_j^2$, all points in the second one should comply with $r_i^2 > r_j^2$.

The idea of the constrained permutation method is to construct L_{ij}^m and L_{ij}^s lines that exist for all pairs of $X_1,...,X_n$ (points) and then consider all corresponding areas in β-space ($\mathbb{M}$ sets) which are created by these lines. Each area represents a single (not necessarily unique) permutation. Now let us have a more detailed look at the constrained permutation LWS method.

Description of the CP-LWS method

Let $X_i = [x_i, y_i]$, i = 1,...,n, be n observations which are assumed to be different ($X_i \neq X_j$ for $i \neq j$) and let $\beta = [\beta_0, \beta_1]$ be a vector in the parameter space for which $y = \beta_0 + \beta_1 x$. Then residuals are given by $r_i(\beta) = y_i - \beta_0 - \beta_1 x_i$.

Definition 4 (L_{ij}^m line). For each two points X_i, X_j $(i < j)$ the midpoint is calculated as $X_{ij}^m = \frac{X_i + X_j}{2}$, where *ij* is an index corresponding to the pair (i, j). The L_{ij}^m line is defined by $\beta_0 = -\beta_1 x_{ij}^m + y_{ij}^m$ and for $[\beta_0, \beta_1] \in L_{ij}^m$: $r_i = -r_j$.

Definition 5 (L_{ij}^s line). For each two points X_i, X_j $(i < j)$ satisfying $x_i \neq x_j$ the slope is evaluated as $s_{ij} = \frac{y_i - y_j}{x_i - x_j}$. The line L_{ij}^s is defined by the slope $\beta_1 = s_{ij}$.

It is obvious that L_{ij}^s lines are constant functions in β-space. For $[\beta_0, \beta_1] \in L_{ij}^s$ the residuals are equal $(r_i = r_j)$. If $x_i = x_j$, the two selected points X_i, X_j give $\beta_1 = \infty$ (see Definition 4) which implies incomputability of the corresponding L_{ij}^s line.

We have two kinds of intersection points in β-space according to their origin, $L_{ij}^m \cap L_{kl}^m$ and $L_{ij}^m \cap L_{kl}^s$. In the first case the parameters corresponding to the intersection point $L_{ij}^m \cap L_{kl}^m$ are calculated as $\beta_0 = \frac{x_{ij}^m y_{kl}^m - x_{kl}^m y_{ij}^m}{x_{ij}^m - x_{kl}^m}$ and $\beta_1 = \frac{y_{ij}^m - y_{kl}^m}{x_{ij}^m - x_{kl}^m}$ if $x_{ij}^m \neq x_{kl}^m$. The L_{ij}^m and L_{kl}^m lines are parallel ($L_{ij}^m \parallel L_{kl}^m$) if $x_{ij}^m = x_{kl}^m$. In the latter case ($L_{ij}^m \cap L_{kl}^s$) we have $\beta_0 = y_{ij}^m - x_{ij}^m \frac{y_k - y_l}{x_k - x_l}$ and $\beta_1 = \frac{y_k - y_l}{x_k - x_l}$ $(x_k \neq x_l)$. These intersections can be viewed as vertices of areas (polygons) in β-space corresponding to a certain permutation of squared residuals. Analysis of squared residuals at the intersection point gives us complete information about all geometrically feasible permutations in adjacent areas. As in the case of P-LWS by inspecting of all intersection points we get also some redundant information (a single area has many vertices), so the same area is inspected multiple times. Since the number of intersection points grows just as n^4 (the number of L_{ij}^m and L_{ij}^s lines ~ n^2), the CP-LWS algorithm should be still much faster than P-LWS with factorial time dependence. Another problem is that the number of possible permutations of squared residuals with the identical values at the intersection point might be actually larger than the number of adjacent areas. In this case the current implementation of the CP-LWS algorithm investigates also permutations that are not geometrically feasible. Later on we will show that this situation arises only if more than two lines accidentally cross each other at the same intersection point and therefore this occurs rarely, especially in a model with stochastic data.

Now we will analyze intersection points involving 4, 3, and 2 mutually different X_i points. This is a very common situation that results from two- or non-accidental three-line crossings of L_{ij}^m or L_{ij}^s lines in β-space.

In the 4-point $L_{ij}^m \cap L_{kl}^m$ intersection case where $r_i^2 = r_j^2$ and $r_k^2 = r_l^2$, we will first analyze mutual relationship between squared residuals. If $r_i^2 < r_k^2$ (at the intersection point), then without loss of generality four areas separated by two lines correspond to 4 different orderings of r^2: $r_i^2 < r_j^2 < r_k^2 < r_l^2$, $r_j^2 < r_i^2 < r_k^2 < r_l^2$, $r_i^2 < r_j^2 < r_l^2 < r_k^2$ and $r_j^2 < r_i^2 < r_l^2 < r_k^2$. These orderings can be represented by 4 permutations (in cycle notation): ( ), (i,j), (k,l) and (i,j)(k,l), where ( ) is the identity permutation (see Figure 4). The last permutation is equal to (k,l)(i,j), because (k,l) and (i,j) are independent. Note that from analysis of squared residuals at the intersection point we cannot obtain information about the actual ordering of residuals in the particular adjacent areas ($\mathbb{M}$ sets). However, this information is not required by CP-LWS algorithm.

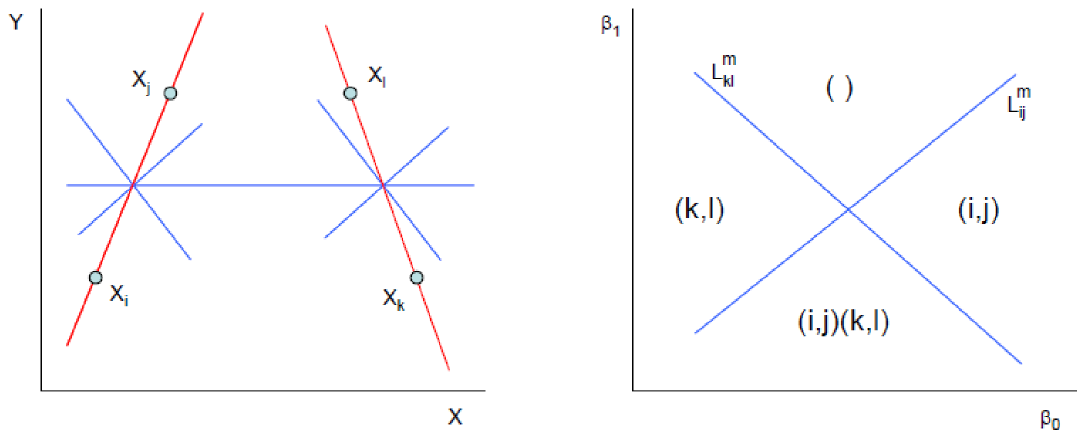

Figure 4
The $L_{ij}^m \cap L_{kl}^m$ intersection in β-space and corresponding orderings of the involved squared residuals represented by 4 permutations in the cycle notation. At the intersection point the residuals satisfy $r_i^2 = r_j^2$ and $r_k^2 = r_l^2$ conditions.

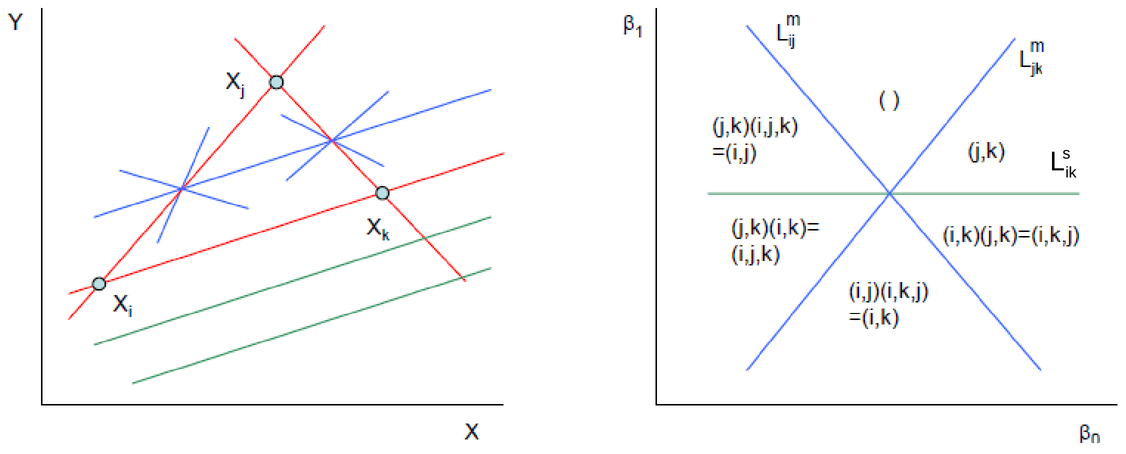

Figure 5
The $L_{ij}^{m} \cap L_{kj}^{m} \cap L_{ik}^{s}$ intersection in β-space and corresponding orderings of the involved squared residuals represented by 6 permutations in the cycle notation. At the intersection point the residuals satisfy the $r_i^2 = r_j^2 = r_k^2$ condition.

At the intersection point involving 3 different points X_i, X_j, X_k all squared residuals equal to each other ($r_i^2 = r_j^2 = r_k^2$). It can be easily shown (see Figure 5) that at this point 3 lines must intersect each other ($L_{ij}^{m} \cap L_{kj}^{m} \cap L_{ik}^{s}$). The orderings of r^2 can be represented by 6 permutations: ( ), (i,j), (i,k), (j,k), (i,j,k) and (i,k,j). The last common case is the intersection point involving 2 different points X_i, X_j ($r_i^2 = r_j^2$) and this point is created by crossing of L_{ij}^{s} and L_{ij}^{m} lines (Figure 3), so we have just 2 permutations: ( ), (i,j).

In the actual implementation of the CP-LWS algorithm, the most common cases of 4, 3, and 2-point intersections generate only geometrically feasible permutations (see Figs. 3-5), and therefore the computer program is running in polynomial time $O\left(n^4\right)$. In the case of accidental degeneracies (*e.g.* when more than 2 lines accidentally cross each other - except the situation described by Figure 5 which is not accidental, or when 2 or more L_{ij}^{m} (L_{ij}^{s}) lines coincide), the number of permutations generated by the program exceeds the number of adjacent areas (geometrically feasible permutations). Thus, the worst-case time complexity of CP-LWS could be comparable with the P-LWS method.

2.2.4 Fast CP-LWS Algorithm

Based on the FastLWS method the Fast CP-LWS algorithm retains all its advantages, namely the superior speed with respect to exact solvers, while it significantly reduces a possibility of ending up in a local LWS minimum. The first step of the Fast CP-LWS algorithm is calculation of the FastLWS estimate. In the second step the CP-LWS algorithm described in Section 2.2.3 is used to search for the global LWS minimum in the vicinity of the FastLWS estimate. Maintaining the superior speed is achieved by selecting only a very limited number of L^m and L^s lines that are closest to the estimated point in β-space. Thus, there are two crucial parameters in Fast CP-LWS: the number of FastLWS evaluations and the number of selected L^m and L^s lines. The first parameter determines the probability of finding a local minimum near the "true" LWS estimate (near hit), and the latter influences the probability of finding the global LWS minimum (exact hit).

The implementation of Fast CP-LWS algorithm will be demonstrated on the following example: for randomly chosen set of 7 points depicted in Figure 6a the FastLWS$_{30}$ and CP-LWS estimates ($c_1 = 0.2, c_0 = 0.45$) are calculated. The corresponding L^m and L^s lines in β-space along with the CP-LWS (in red) and FastLWS (in green) solutions are visualized in Figure 6b. Note that in this particular example the FastLWS and CP-LWS solutions nearly

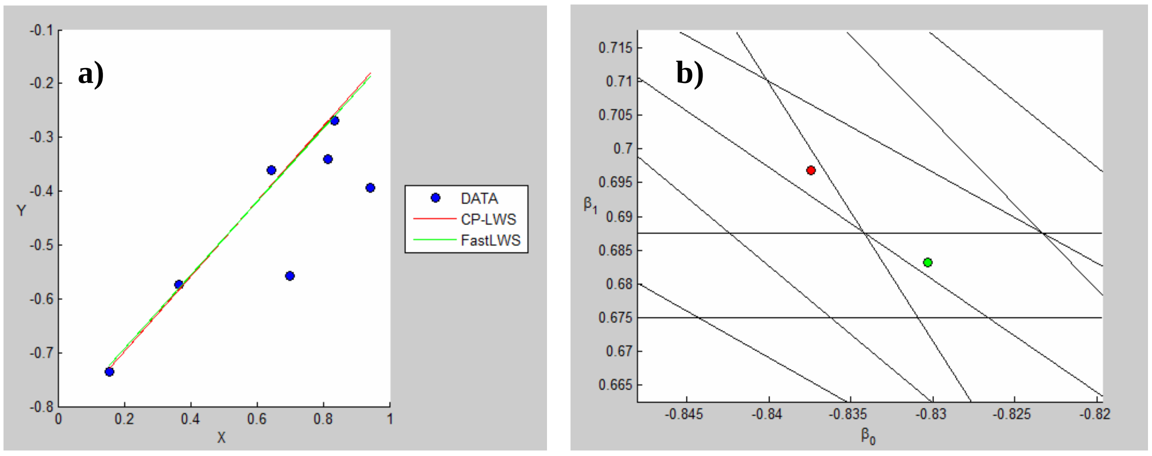

Figure 6
Example of the Fast CP-LWS algorithm: a) for a given set of points (blue) the LWS solution and its FastLWS approximation is calculated and b) results are shown in β-space along with the nearest L^m and L^s (horizontal) lines.

coincide (Figure 6a) since these solutions are found in adjacent areas in β-space (Figure 6b). The Fast CP-LWS algorithm guarantees finding the exact LWS solution provided that at least one of vertices of the polygon containing the $\hat{\beta}^{LWS}$ (red point in Figure 6b) is created by selected L^m and L^s lines.

2.2.5 Differential Evolution LWS Algorithm

As an alternative to the FastLWS and Fast CP-LWS algorithms we have implemented also a direct minimization of the LWS objective function (2.10) in β-space using a stochastic, population-based optimization algorithm introduced by Storn and Price [Storn 1996] known as Differential Evolution (DE). Nowadays DE has become one of the most frequently employed evolutionary algorithms (genetic algorithms, evolutionary strategies) to deal with the global optimization problems. The DE algorithm is particularly well suited for solving the LWS problem (2.8) – it has been developed to optimize a real parameter, real valued function and offers a way of optimizing such a function without evaluating its gradient. This is useful for our purpose since the LWS objective function has many discontinuities in the first derivative and consequently the gradient-based optimization methods cannot be applied. Moreover, the DE algorithm has been shown to perform outstandingly even for very "hard" optimization problems involving multiple minima functions with nearly degenerate values of the cost (objective) function at these minima. Again, this is precisely the reason why the LWS problem (2.8) is so hard to tackle with ordinary optimization methods. Standard deterministic algorithms tend to stop the minimization in a local LWS minimum nearest to the initial starting point. The last but not least, the DE algorithm is very easy to implement. The actual MATLAB code performing the DE minimization of (2.8) consists just of 20 lines.

Let us first introduce basic components of the DE algorithm. Consider an objective (cost, error, fitness) function $f:\mathbb{R}^D \to \mathbb{R}$. The population $P_{\mathbf{x},G} = \{\mathbf{x}_{0,G}, \mathbf{x}_{2,G}, \ldots, \mathbf{x}_{N-1,G}\}$ is a collection of trial parameter vectors, $\mathbf{x}_{i,G}$, G is the generation number and N is the population size. Each of the N parameter vectors undergoes mutation, recombination and selection process resulting in overall improvement of f in the next generation. The classical variant of DE denoted in literature by abbreviation DE/rand/1/bin consists of the following steps (see classical DE flowchart depicted in Figure 7):

1) **Initialization**

The members of the initial generation $P_{\mathbf{x},0}$ are chosen randomly (typically the upper and lower bounds for each component of the parameter vector are specified and the initial parameter values are generated uniformly on these intervals).

2) **Mutation**

The purpose of the mutation step is to expand the search space. For a given parameter vector $\mathbf{x}_{i,G}$ there are three randomly selected vectors $\mathbf{x}_{r1,G}, \mathbf{x}_{r2,G}, \mathbf{x}_{r3,G}$ such that the indices *i*, *r*$_1$, *r*$_2$ and *r*$_3$ are mutually distinct. A mutant vector $\mathbf{v}_{i,G}$ is generated by adding the weighted difference of two of the vectors to the third

$$\mathbf{v}_{i,G} = \mathbf{x}_{r1,G} + F\left(\mathbf{x}_{r2,G} - \mathbf{x}_{r3,G}\right), \tag{2.11}$$

where $F > 0$ is an input parameter controlling the mutation step.

3) **Recombination**

Recombination (or crossover) incorporates successful solutions from the previous generation. The purpose of the recombination step is the diversity enhancement. The trial vector $\mathbf{u}_{i,G} = (u_{i,G}^1, \ldots, u_{i,G}^D)$ is constructed from the elements of the target vector, $\mathbf{x}_{i,G}$, and the elements of the mutant vector, $\mathbf{v}_{i,G}$. Elements of the mutant vector enter the trial vector with probability $CR \in [0,1]$ (input parameter controlling the recombination step):

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } U_j \le CR \quad \text{or} \quad j = l \\ x_{i,G}^j & \text{if } U_j > CR \quad \text{and} \quad j \ne l, \end{cases} \tag{2.12}$$

where *l* is randomly chosen integer from $1, 2, \ldots, D$ and $U_1, U_2, \ldots, U_D$ are random variables uniformly distributed in $[0,1)$. Eq. (2.12) ensures that at least one element of the target vector is replaced by the corresponding element of the mutation vector. The recombination process described above is called the binomial crossover (the second commonly used type of recombination process is the exponential crossover, see [Price 2005] for more details).

4) **Selection**

The target vector $\mathbf{x}_{i,G}$ is compared with the trial vector $\mathbf{u}_{i,G}$ and the one with the lowest objective function value is selected to the next generation (tournament selection with one-to-one survivor selection).

5) Termination

Mutation, recombination and selection steps (2-4) continue until some kind of stopping criterion is reached and the DE algorithm terminates.

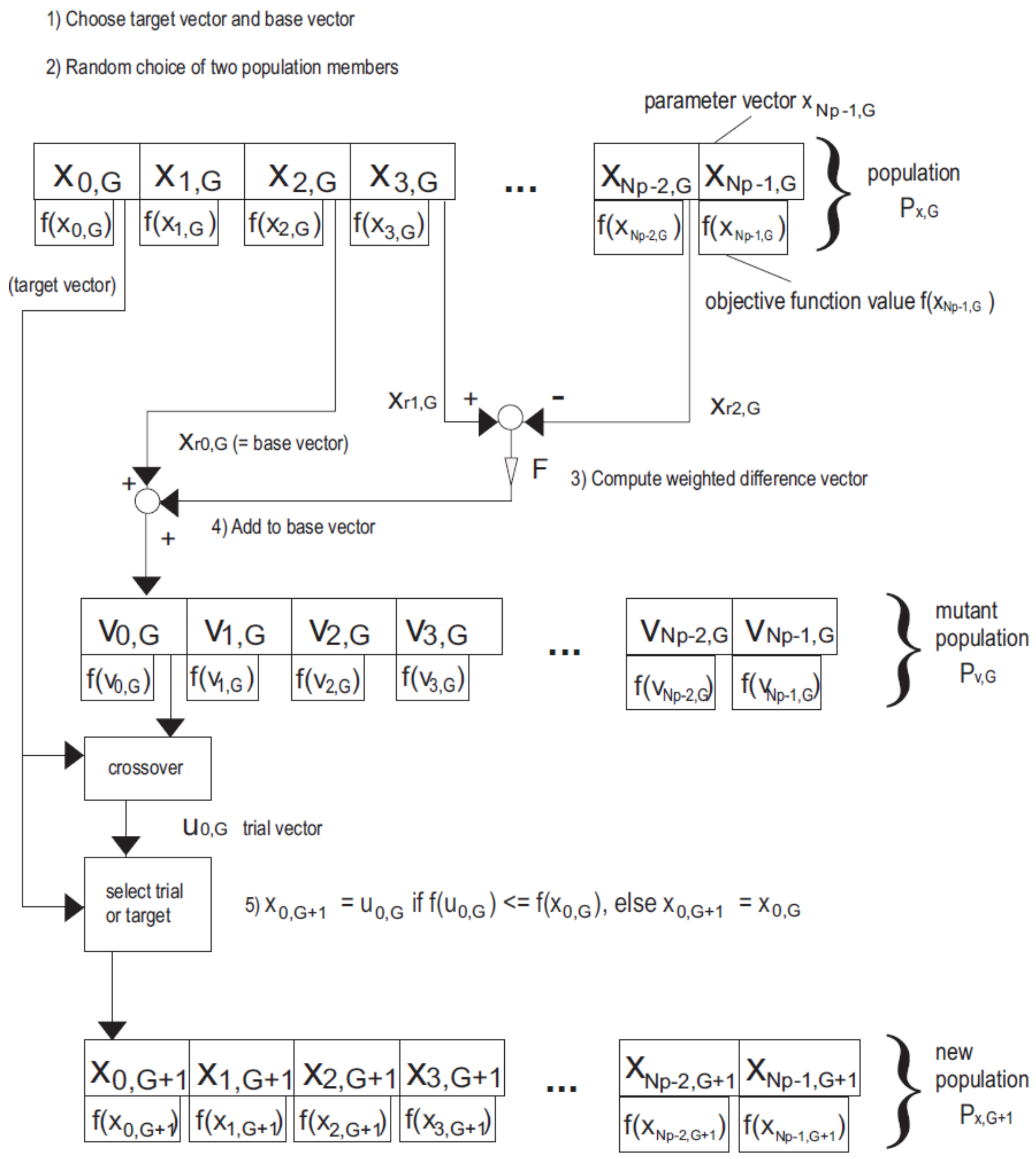

Figure 7
The flowchart of classical Differential Evolution.

There are a few control parameters in differential evolution, namely the size of population (*N*), the parameter controlling the mutation process (*F*), and the crossover parameter (*CR*) controlling the diversity enhancement. The efficiency of DE is particularly sensitive to the settings of parameters *F* and *CR*, so the optimal values of the control parameters has to be found by trial-and-error tuning (although self-adaptation of control parameters has been proposed, see [Brest 2006]). The rule of thumb values for the control variables have been given by [Storn 1997]: $F \in [0.5, 1.0]$, $CR \in [0.8, 1.0]$, and $N = 10 \cdot D$.

In our implementation of the classical DE/rand/1/bin algorithm (rand, 1, and bin stand for random selection of the base vector $\mathbf{x}_{r1,G}$, number of difference vectors used in mutation step (2.11), and the type of crossover (2.12), respectively), which has been tailored for the LWS optimization problem (2.8), we have adopted several modifications of the procedure described above: starting vectors in $P_{\mathbf{x},0}$ (initialization step) were obtained by random sampling of β-space generated by the ordinary least squares (OLS) estimator for selected subset of *p* observations (this sampling was used also for the FastLWS algorithm, details are given in Section 2.2.2). The size of population of 100 (*N*) was found to be sufficient for all datasets used in this work. The control parameters are set to 0.8 and 0.7 for *F* and *CR*, respectively. This setting has been found optimal for the $D = 2$ case. Regression models with more β coefficients ($p > 2$) might require a somewhat larger value of *CR*. Note that the average number of replacements in (2.12) depends on *D* due to the requirement that at least one element of the target vector must be changed by the binomial crossover. A simple termination criterion (step 5) limiting the number of generations (G < 200) is used. This is probably a very conservative setting since the DE-LWS algorithm converges to the global LWS minimum typically in less than 100 iterations. Finally, the calculated DE estimate of $\hat{\beta}^{LWS}$ (β^{DE}) enters the FastLWS$_1$ evaluation to obtain a numerically exact solution of (2.8). The FastLWS correction guarantees to obtain the LWS global minimum ($\hat{\beta}^{LWS}$) provided that both β^{DE} and $\hat{\beta}^{LWS}$ are within the interior of the same $\mathbb{M}$ set (see Section 2.2.3).

3. Comparison of LWS Algorithms

In this part we compare MATLAB implementations of the LWS algorithms described in Section 2 from several points of view, the computational speed and reliability of inexact methods in particular. We investigate the behavior of our implementations in various situations using random data with different number of points, varying the standard deviation of disturbances, different number of outliers and coefficient of contamination (percentage of data generated by a second regression model). Apart from the model parameters (details about data modeling can be found in Section 1.2) the particular form of the weight function ψ introduced in Section 2.1 need to be specified. Let us recall that ψ is a continuous non-increasing function $\psi : [0,1] \rightarrow [0,1]$ such that $\psi(0)=1$ and $\psi(1)=0$. The corresponding weights are calculated as $w_i = \psi\left(\frac{i-1}{n}\right), \quad i=1,\ldots,n$. A choice of the particular weight function is arbitrary and many examples of convenient functional forms can be found in literature. In this work we have employed a piecewise linear function consisting of three linear segments depicted in Figure 8. The weight function has only two parameters (c_1, c_0) determining the length of the first ($\psi = 1$) and third ($\psi = 0$) segment

$$\psi(x) = \begin{cases} 1 & \text{if} \quad x \le c_1 \\ \dfrac{x + c_0 - 1}{c_1 + c_0 - 1} & \text{if} \quad c_1 < x \le 1 - c_0 \\ 0 & \text{if} \quad x > 1 - c_0 \end{cases} \tag{3.1}$$

The weight function (3.1) is evaluated at the discrete set of points $x_1, x_2, \ldots, x_n \in [0,1)$ given by $x_k = \frac{k-1}{n}, \quad k = 1,\ldots n$ (red points in Figure 8). Our choice of weight function combines both simplicity and sufficient flexibility. It covers all important limiting cases of LWS: $c_1 = 1, c_0 = 0$ ($w_i = 1, \quad i = 1,\ldots,n$) corresponds to the standard ordinary least squares (OLS) estimator $\hat{\beta}^{LWS} = \hat{\beta}^{OLS}$. The $c_1 = 1 - c_0$ setting ($w_i = 1, \quad i = 1,\ldots,l$ and $w_i = 0, \quad i = l+1,\ldots,n$, where $l = \operatorname{int}(n \cdot c_1 + 1)$ is an integer number rounded toward zero) corresponds to the least trimmed squares (LTS) estimator $\hat{\beta}^{LWS} = \hat{\beta}^{LTS}$. The parametr c_0 controls the breakdown point of the LWS estimator, *i.e.* the percentage of "bad" data (type (i) or (ii) of data contamination discussed in Section 1.2) at which the estimator breaks down. Maximum breakdown point is 50% which means that any majority can overrule any minority, as it is in the case of the sample median [Hampel 1986]. Hence, the largest reasonable value for c_0 parameter is 0.5. The parameter c_1 can be chosen arbitrarily at the interval $[0, 1-c_0]$. Note that for any setting of c_1 at least one point has weight equal to 1.

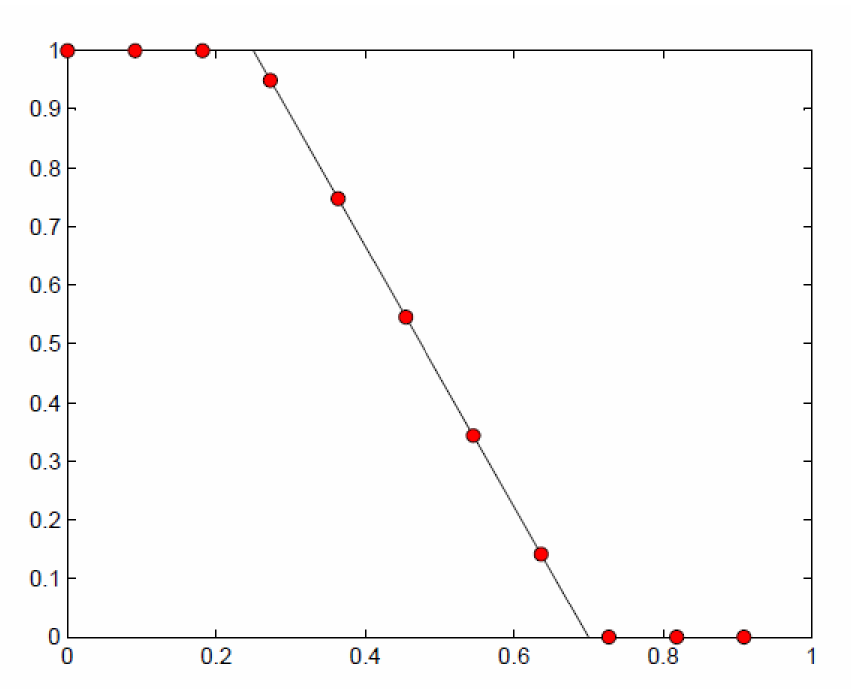

Figure 8
The weight function constructed as a piecewise linear function with $n = 11,\ c_1 = 0.25,\ c_0 = 0.3$.

3.1 <u>Performance of Exact LWS Methods</u>

The CPU time requirements of P-LWS and CP-LWS methods are shown in Figure 9. The comparison was caried out for very small datasets (n < 10) only due to severe speed limitations of the P-LWS algorithm. Considering time complexity of $O(n!)$, the P-LWS algorithm is useful perhaps for testing computationally more efficient methods such as CP-LWS proposed in this work. For actual benchmarks involving thousands of LWS evaluations the P-LWS algorithms is clearly inadequate.

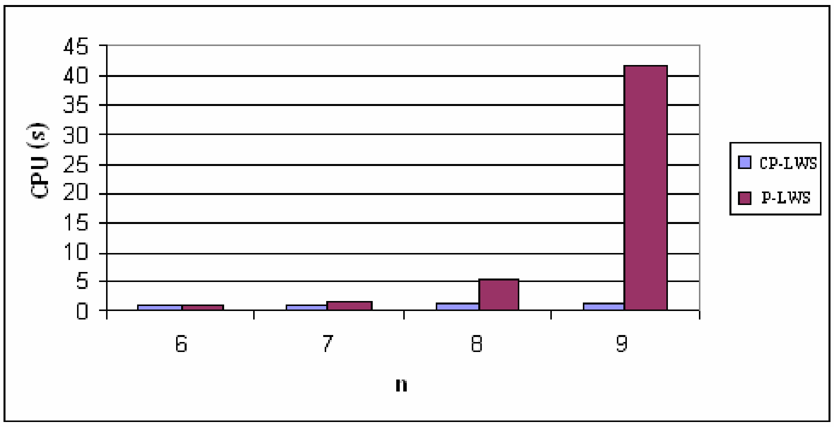

Figure 9
Comparison of the P-LWS and CP-LWS algorithms.

3.2 Performance of Approximative LWS Methods

Due to the superior speed of inexact LWS methods the CPU time demands are of less concern. Thus, we focus primarily on accuracy and reliability of these methods. The actual tests are performed using CP-LWS (exact) algorithm which is fast enough to calculate several thousands of LWS evaluations even for moderately large datasets ($n < 20$).

3.2.1 Assessment of Reliability of the FastLWS Method

To assess the reliability of the FastLWS method we have carried out exactness benchmark test proposed in [Skuhrovec 2009] using Fast LWS$_{30}$ (30 FastLWS evaluations) and CP-LWS calculations for 1000 randomly generated datasets (see Section 1.2 for definition of the model parameters). The results are summarized in Tables 1-4. We report the number of exact hits ($\hat{\beta}^{FastLWS} = \hat{\beta}^{LWS}$) and statistics (mean, max, var) on the near hits ($\hat{\beta}^{FastLWS} \neq \hat{\beta}^{LWS}$) evaluated for the difference between the FastLWS and CP-LWS (exact) objective functions

$$S^2(\beta, h) = \sum_{l=1}^{n} \psi\left(\frac{h_l - 1}{n}\right) r_l^2(\beta).$$

Table 1 Exactness benchmark of FastLWS$_{30}$ for various dataset sizes (*n*).

Model					Number of exact hits (out of 1000)	Statistics on near hits		
n	sigma	n_o	c_1	c_0		Mean S^2 deviation	Max S^2 deviation	Var S^2 deviation
5	0.1	0	0.0	0.0	971	4.0E-05	4.6E-04	8.7E-09
7	0.1	0	0.0	0.0	963	6.1E-05	6.6E-04	1.8E-08
9	0.1	0	0.0	0.0	962	7.2E-05	9.2E-04	2.7E-08
11	0.1	0	0.0	0.0	955	9.9E-06	4.3E-05	1.2E-10
13	0.1	0	0.0	0.0	945	1.6E-05	1.1E-04	4.7E-10
15	0.1	0	0.0	0.0	941	1.5E-05	3.7E-04	2.4E-09
17	0.1	0	0.0	0.0	938	1.3E-05	1.8E-04	8.1E-10
19	0.1	0	0.0	0.0	917	2.9E-06	2.7E-05	2.2E-11

Table 2 Exactness benchmark of FastLWS$_{30}$ for models with various standard deviations of disturbances (homoskedastic model).

Model					Number of exact hits (out of 1000)	Statistics on near hits		
n	sigma	n_o	c_1	c_0		Mean S^2 deviation	Max S^2 deviation	Var S^2 deviation
11	**0.1**	0	0.0	0.0	955	9.9E-06	4.3E-05	1.2E-10
11	**0.2**	0	0.0	0.0	954	3.5E-05	1.7E-04	1.7E-09
11	**0.3**	0	0.0	0.0	955	6.9E-05	3.7E-04	6.3E-09
11	**0.4**	0	0.0	0.0	953	1.5E-04	6.9E-04	3.0E-08
11	**0.5**	0	0.0	0.0	956	2.3E-04	1.1E-03	7.7E-08

Table 3 Exactness benchmark of FastLWS$_{30}$ for various number of outliers (n_o).

Model					Number of exact hits (out of 1000)	Statistics on near hits		
n	sigma	n_o	c_1	c_0		Mean S^2 deviation	Max S^2 deviation	Var S^2 deviation
11	0.1	**6**	0.0	0.5	871	4.0E-02	9.7E-01	1.1E-02
11	0.1	**5**	0.0	0.5	895	1.8E-04	2.9E-03	1.7E-07
11	0.1	**4**	0.0	0.5	864	9.2E-05	1.8E-03	3.9E-08
11	0.1	**3**	0.0	0.5	847	8.3E-05	1.6E-03	3.9E-08
11	0.1	**2**	0.0	0.5	862	4.0E-05	4.5E-04	6.0E-09
11	0.1	**1**	0.0	0.5	838	2.2E-05	3.3E-04	2.1E-09
11	0.1	**0**	0.0	0.5	828	2.5E-05	8.5E-04	6.4E-09

The exactness benchmarks show rather stable performance of FastLWS with respect to the number of points *n* and standard deviation of disturbances (Tables 1 and 2). We observe a small decrease of the number of exact hits with increasing number of points, the number of hits remains practicaly constant with increasing σ. The results in Table 3 show dependence of FastLWS performance on the number of outliers for the highest breakdown point settings ($c_0 = 0.5$) of the associated weight function. FastLWS performs reasonably well even if the number of exact hits is significantly lower than for $c_0 = 0$. The statistics on near hits exhibit large increase for models close to the breakdown point ($n_o \geq 5$). Largest variations in the number of exact hits and the statistics on near hits are shown in Table 4 where these statistics are calculated for various values of the weight function parameters. Interestingly, the best performance of FastLWS is observed for c_1, c_0 parameters corresponding to the least trimmed squares (LTS) setting (the last row in Table 4).

Table 4 Exactness benchmark of FastLWS$_{30}$ for various parameters of the weight function.

Model					Number of exact hits (out of 1000)	Statistics on near hits		
n	sigma	n_o	c_1[a]	c_0[a]		Mean S^2 deviation	Max S^2 deviation	Var S^2 deviation
7	0.1	2	**0.00 (1)**	**0.00 (0)**	975	7.0E-02	3.7E-01	1.3E-02
7	0.1	2	**0.00 (1)**	**0.15 (1)**	965	7.9E-02	5.8E-01	2.3E-02
7	0.1	2	**0.00 (1)**	**0.29 (2)**	962	1.0E-04	1.1E-03	3.6E-08
7	0.1	2	**0.00 (1)**	**0.43 (3)**	886	8.3E-05	1.4E-03	4.4E-08
7	0.1	2	**0.15 (2)**	**0.00 (0)**	977	9.9E-02	5.1E-01	2.5E-02
7	0.1	2	**0.15 (2)**	**0.15 (1)**	972	1.2E-01	7.0E-01	4.0E-02
7	0.1	2	**0.15 (2)**	**0.29 (2)**	978	1.8E-04	1.3E-03	8.5E-08
7	0.1	2	**0.15 (2)**	**0.43 (3)**	940	2.0E-04	2.8E-03	2.0E-07
7	0.1	2	**0.29 (3)**	**0.00 (0)**	986	1.7E-01	5.1E-01	3.7E-02
7	0.1	2	**0.29 (3)**	**0.15 (1)**	976	1.7E-01	8.7E-01	6.5E-02
7	0.1	2	**0.29 (3)**	**0.29 (2)**	982	1.6E-04	5.5E-04	3.0E-08
7	0.1	2	**0.29 (3)**	**0.43 (3)**	980	6.3E-04	3.8E-03	9.1E-07
7	0.1	2	**0.43 (4)**	**0.00 (0)**	985	2.0E-01	6.1E-01	4.4E-02
7	0.1	2	**0.43 (4)**	**0.15 (1)**	985	3.1E-01	1.1E+00	1.1E-01
7	0.1	2	**0.43 (4)**	**0.29 (2)**	994	2.8E-04	7.0E-04	4.7E-08
7	0.1	2	**0.43 (4)**	**0.43 (3)**	995	4.8E-03	1.1E-02	1.7E-05

[a]The number in parentheses is a total number of points with $\psi = 1$ and $\psi = 0$ for c_1 and c_0, respectively.

The results reported in Tables 1-4 can be summarized as follows: the performance of the FastLWS method measured by a ratio of exact hits ($\hat{\beta}^{FastLWS} = \hat{\beta}^{LWS}$) to the total number of trials on randomly generated data samples (denoted as success probability of FastLWS) is rather insensitive to the model parameters such as number of points, standard deviation of disturbances or number of outliers. This behavior is expected to be preserved also for larger datasets (n>15). Observed strong dependence of the FastLWS success probability on the weight function parameters, however, deserves a closer examination. Toward this end, we have evaluated two additional statistics by the CP-LWS algorithm – the average number of LWS minima and the fraction of the favorable permutations leading to the global LWS minimum ($\hat{\beta}^{LWS}$) in the FastLWS optimization process. Results plotted in Figure 10 are a bit surprising. First, the average number of LWS minima and the fraction of favorable permutations correlates with the FastLWS success probability only for points with the same value of c_0 (the same number of points with zero weight). Second, the theoretical success probability (black curve in Figure 10b) calculated for 30 FastLWS evaluations as $1-(1-P_f)^{30}$, where P_f is the fraction of favorable permutations (probability of finding $\hat{\beta}^{LWS}$ in a single FastLWS evaluation starting from a randomly chosen feasible permutation), clearly indicates highly inefficient sampling of β-space. Assuming uniform sampling of feasible permutations (sample permutations being drawn with equal probability), the success rate for 30 FastLWS evaluations should be very close to 1. At this point let us recall that our implementation of the FastLWS algorithm uses sampling of parameter space generated by the ordinary least squares (OLS) estimator for randomly chosen subset of p observations. In this particular case (n=7, p=2) the complete sampling space includes β generated from 21 distinct pairs of points, most of them are explored in 30 FastLWS evaluations. This means that the results in Figure 10 are nearly converged with respect to the number of the FastLWS evaluations. Obviously, the observed failures of the FastLWS algorithm can be related to the implemented sampling method (in some cases no OLS sample permutation leads to $\hat{\beta}^{LWS}$ in FastLWS optimization). To verify this conclusion, we have carried out the same test for full sampling of parameter space generated by the OLS estimator for complete subset of p = 2 observations, *i.e.* for 21 different pairs.

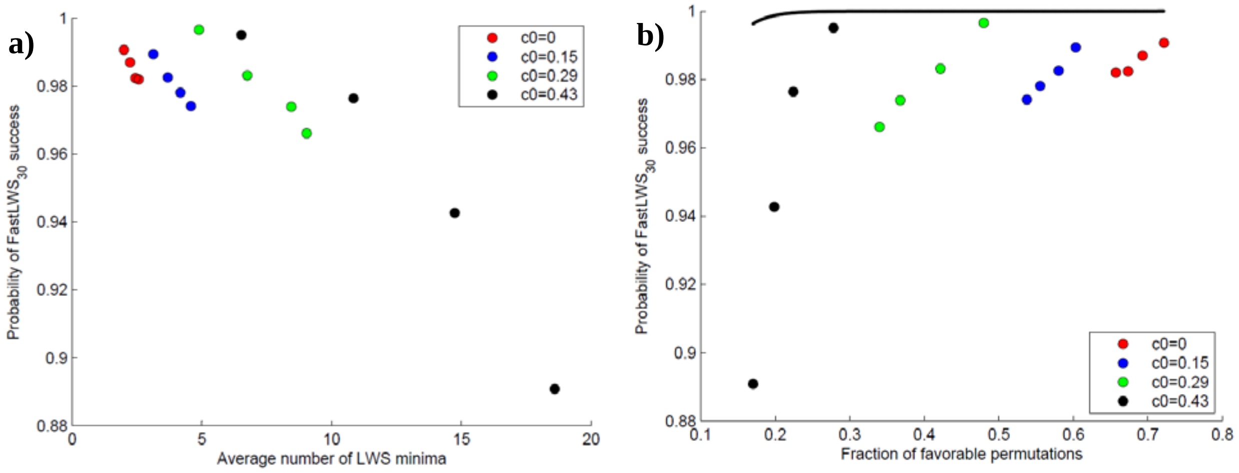

Figure 10
Success rate of the FastLWS algorithm with respect to a) average number of LWS minima and b) fraction of favorable permutations. Evaluated for 10000 random datasets and different weight function parameters (c_1=0, 0.15, 0.29, 0.43 for each value of c_0). The black curve represents predicted success probability assuming uniform sampling of feasible permutations.

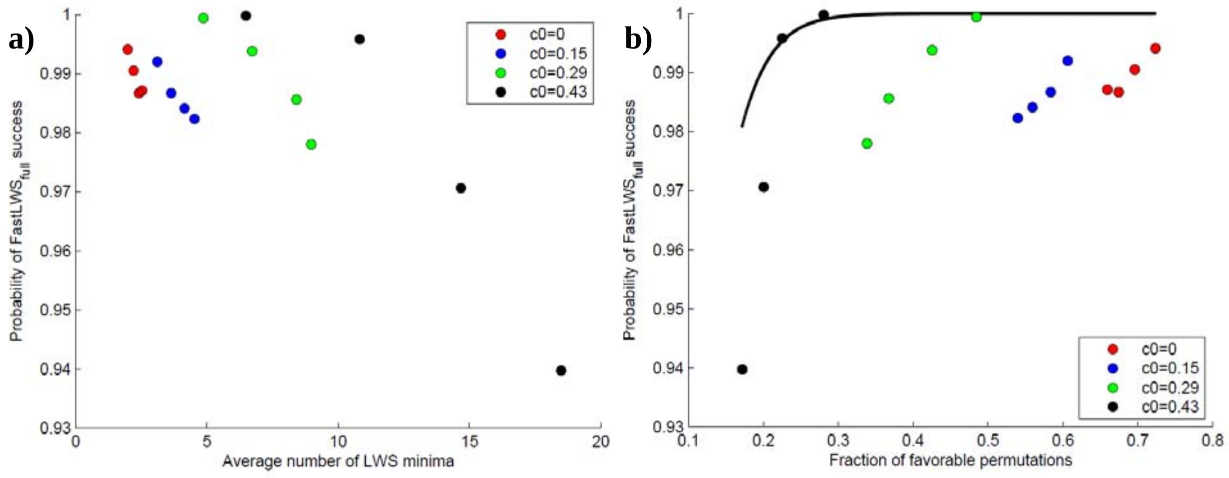

Figure 11
Success rate of the FastLWS algorithm (using the complete OLS sampling) with respect to a) average number of LWS minima and b) fraction of favorable permutations.
Evaluated for 10000 random datasets and different weight function parameters (c_1=0, 0.15, 0.29, 0.43 for each value of c_0). The black curve represents predicted success probability assuming uniform sampling of feasible permutations.

The results are shown in Figure 11. As expected, the overall picture remains unchanged confirming thus conclusions drawn from the previous test. The predicted probability of FastLWS success using the complete OLS sampling is calculated as $1-(1-P_f)^{21}$. The FastLWS results obtained with weights corresponding to the LTS estimator ($c_1 = 0.43, c_0 = 0.43$) or similar settings ($c_1 = 0.29, c_0 = 0.43$ and $c_1 = 0.43, c_0 = 0.29$) show the best performance and nicely fit the predicted values. The most difficult case for the FastLWS algorithm seems to be $c_1 = 0, c_0 = 0.43$ yielding weights *w*=1, 0.75, 0.5, 0.25, 0, 0, 0. This setting produces the largest number of LWS minima and consequently the lowest fraction of favorable permutations. Thus, we can conclude that OLS sampling of β-space is certainly a good choice for the LTS estimator. For a general LWS estimator, however, a better sampling method should be devised. The quest for a better sampling method actually motivated our attempt to employ a direct optimization in β-space. We will discuss the stochastic optimization strategy in Section 3.2.3.

3.2.2 Assessment of Reliability of the Fast CP-LWS Method

The same exactness benchmark for various parameters of the weight function (Table 4 and Figures 10-11) was carried out with the Fast CP-LWS algorithm proposed in Section 2.2.4.

The FastLWS$_{30}$ calculations were augmented with CP-LWS search in β-space using L^m and L^s lines sorted by the distance from the FastLWS estimate $\beta^{FastLWS}$ (defined geometrically in 2-dimensional Euclidean space). Our main goal is to obtain information about an effective distance (defined topologically) of FastLWS estimate from $\hat{\beta}^{LWS}$ measured by the lowest number of the nearest L^m and L^s lines required to calculate the global LWS minimum. This information is crucial for assessment of reliability of the Fast CP-LWS algorithm since it relies to large extent on FastLWS. Note that unrestricted CP-LWS search in the space of all feasible permutations must finally end up in the global LWS minimum. In practice, however, we have to restrict the search by a maximum number of L^m and L^s lines selected for the CP-LWS algorithm. Thus, we will always sacrifice reliability (exactness) of CP-LWS to computational efficiency. If the FastLWS (local minimum) solution is very far form $\hat{\beta}^{LWS}$, CP-LWS calculation will necessarily fail. Table 5 summarizes the average and maximum number of L^m and L^s lines required to obtain $\hat{\beta}^{LWS}$ (global minimum) evaluated for 1000 and 10000 random datasets with different weight function parameters.

Table 5 Performance of FastLWS$_{30}$ for various parameters of the weight function.

Model					1 000 trials			10 000 trials		
n	sigma	n_o	c_1[a]	c_0[a]	Number of exact hits	Average FastLWS distance	Max. FastLWS distance	Number of exact hits	Average FastLWS distance	Max. FastLWS distance
7	0.1	2	**0.00 (1)**	**0.00 (0)**	975	2.56	7	982	2.59	10
7	0.1	2	**0.00 (1)**	**0.15 (1)**	965	3.77	11	974	3.40	11
7	0.1	2	**0.00 (1)**	**0.29 (2)**	962	2.18	5	966	2.17	5
7	0.1	2	**0.00 (1)**	**0.43 (3)**	886	2.20	7	891	2.18	8
7	0.1	2	**0.15 (2)**	**0.00 (0)**	977	2.35	7	982	2.60	11
7	0.1	2	**0.15 (2)**	**0.15 (1)**	972	4.21	11	978	3.69	11
7	0.1	2	**0.15 (2)**	**0.29 (2)**	978	2.41	5	974	2.23	5
7	0.1	2	**0.15 (2)**	**0.43 (3)**	940	2.43	7	943	2.35	8
7	0.1	2	**0.29 (3)**	**0.00 (0)**	986	2.64	7	987	2.71	8
7	0.1	2	**0.29 (3)**	**0.15 (1)**	976	3.83	10	983	3.67	11
7	0.1	2	**0.29 (3)**	**0.29 (2)**	982	2.33	5	983	2.28	5
7	0.1	2	**0.29 (3)**	**0.43 (3)**	980	3.10	7	976	2.74	8
7	0.1	2	**0.43 (4)**	**0.00 (0)**	985	2.93	7	991	2.99	9
7	0.1	2	**0.43 (4)**	**0.15 (1)**	985	4.67	10	989	4.79	16
7	0.1	2	**0.43 (4)**	**0.29 (2)**	994	2.50	4	997	2.59	6
7	0.1	2	**0.43 (4)**	**0.43 (3)**	995	4.60	7	995	4.33	8

[a]The number in parentheses is a total number of points with $\psi = 1$ and $\psi = 0$ for c_1 and c_0, respectively.

The results suggest that the average and maximum FastLWS distance from $\hat{\beta}^{LWS}$ does not depend on the FastLWS success rate. The most problematic case ($c_1 = 0, c_0 = 0.43$) with the lowest success probability has even lower average distance (2.2 lines) than the most reliable LTS setting ($c_1 = 0.43, c_0 = 0.43$) with the average value of 4.3 lines. On the other hand, Table 5 shows that the Fast CP-LWS calculation with just 11 selected L^m and L^s lines gives the $\hat{\beta}^{LWS}$ estimate with success probability of 99.9% or better for all weight function settings (cf. with 89% for FastLWS$_{30}$).

3.2.3 Comparison of the FastLWS, Fast CP-LWS, and DE-LWS Methods

In this Section we will compare inexact LWS algorithms based on discrete search in the space of all geometrically feasible permutations, $\mathbb{H}_f$, with stochastic optimization in β-space using the Differential Evolution (DE) algorithm described in 2.2.5. The comparison is performed on moderately large datasets consisting of 50 points drawn from two different regression models (type (ii) of data contamination) with coefficient of contamination of 0.2 (20% of data comes from the second regression model). The most difficult setting of weigth function parameters ($c_1 = 0, c_0 = 0.5$) with the lowest success rate is used. Since the exact CP-LWS calculation takes already a few minutes on datasets of this size, we have to rely on comparison of solutions obtained by inexact algorithms (Fast CP-LWS and DE-LWS) with conservative settings of control parameters and the lowest calculated value of the LWS objective function is taken as a global minimum. Both methods have success probability better than 99.9% for smaller datasets (<20), where the exact CP-LWS calculations are possible for ~10^5 LWS evaluations. Note that the computationally most efficient FastLWS algorithm is not suitable for the purpose of this test due to very low success prabability. A brief inspection of Table 6 reveals that the DE-LWS method is superior in both computational speed and reliability. In fact, we used DE-LWS results as a reference for performance evaluation of permutation-based inexact methods. Table 6 also shows that the Fast CP-LWS algorithm significantly improves FastLWS reliability. The most important feature of the Fast CP-LWS method is a possibility of systematic improvement of success probability even though at the cost of computational efficiency. To obtain 100% success rate 50 nearest L^m and L^s lines are required for CP-LWS search which adds about 0.65s of CPU time to the FastLWS computation with 250 or 500 evaluations. Comparing to DE-LWS it represents an increase in relative performance by 3.4

(it takes 3.4 times more CPU time than DE-LWS). As mentioned above, performance of Fast CP-LWS algorithm depends strongly on accuracy of $\hat{\beta}^{LWS}$ estimate provided by FastLWS. Unfortunately, FastLWS is not particularly efficient from this point of view (see FastLWS results in Table 6) mainly due to the employed OLS sampling of β-space. Thus, combined with more reliable method for calculating estimate of $\hat{\beta}^{LWS}$, the CP-LWS search implemented in Fast CP-LWS might easily outperform the DE-LWS algorithm.

Table 6 Comparison of the FastLWS, CP-LWS and DE-LWS methods.[a]

Method	FastLWS Evaluations	Number of L^m and L^s lines	Success Probability	CPU Time [s]	Relative[b] Performance
FastLWS	30	-	0.726	0.04	0.2
	100	-	0.829	0.14	0.7
	250	-	0.892	0.34	1.8
	500	-	0.917	0.68	3.6
Fast CP-LWS	30	5	0.896	0.05	0.3
	30	10	0.931	0.07	0.4
	30	25	0.974	0.21	1.1
	30	50	0.988	0.71	3.7
	100	5	0.955	0.15	0.8
	100	10	0.972	0.16	0.8
	100	25	0.992	0.30	1.6
	100	50	0.999	0.79	4.2
	250	5	0.974	0.35	1.8
	250	10	0.989	0.37	1.9
	250	25	0.999	0.50	2.6
	250	50	1.000	0.99	5.2
	500	5	0.983	0.69	3.6
	500	10	0.992	0.71	3.7
	500	25	0.998	0.85	4.5
	500	50	1.000	1.34	7.1
DE-LWS	-	-	1.000	0.19	1.0

[a]Calculated for 1000 randomly chosen regression models with 50 observations and the coefficient of contamination of 0.2. The weight function parameters were set to $c_1 = 0, c_0 = 0.5$.

[b]Relative to DE-LWS.

4. LWS Estimation of the Heteroscedastic Regression Model

Regression model (1.1) with disturbances having non-constant variances across observations

$$\operatorname{var}(e_i) = E[(e_i - E(e_i))^2] = E(e_i^2) = \sigma_i^2 , \tag{4.1}$$

is called heteroscedastic. The disturbances are assumed to be uncorrelated. Let us recall that under assumption of homoscedasticity (and other basic assumptions given in Section 1.1) $\hat{\beta}^{(OLS,n)}$ is the best linear unbiased estimator (BLUE), *i.e.* it has minimum variance among all linear unbiased estimators. In the presence of heteroscedasticity the OLS estimator is still unbiased, consistent, and asymptotically normally distributed, but it is inefficient with respect to the Generalized Least Square (GLS) estimator [Greene 2003]. It means that we can still use the OLS results even if we do not know the precise nature of the heteroscedasticity (if the form of heteroscedasticity is known, the use of the GLS esimator is preferable [Greene 2003]). The OLS results may not be optimal, but at least they are not completely wrong. The presence of heteroscedasticity has, however, some (potentially dangerous) implications for inferences based on OLS. Therefore, it is highly desirable to be able to test for the presence of heteroscedasticity. Several tests have been proposed, most of them are based on consistency of the OLS estimator. In this work we will focus on the test proposed by [White 1980].

4.1 White's Heteroscedasticity Test

The main idea of White test [White 1980] is to compare elements of two different estimation of matrix $\frac{1}{n}\sigma^2 X^T X$:

$$\frac{1}{n} X^T X \frac{\sum_{i=1}^{n} r_i^2}{n} \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^{n} X_i X_i^T r_i^2 , \tag{4.2}$$

where r_i^2 are squared residuals and X_i is the i-th row of $\boldsymbol{X}$. In the presence of homoscedasticity of disturbances, estimates (4.2) should give the same results, in the presence of heteroscedasticity these two estimates will diverge. The White's test is not particularly powerful, but it can be used in situations when one has little idea if there exists heteroscedasticity, and no idea at all of its potential form.

A practical implementation of the test was proposed by [White 1980] (under the assumption that disturbances are homokurtic ($E\left(e_i^4\right) = \mu_4$ for all i) and the simplified test proceeds as follows:

From original regression model squared residual are estimated and the following (auxiliary) regression is constructed

$$r_i^2 = \alpha_0 + \sum_{j=1}^{p}\sum_{k=j}^{p} \alpha_s X_{ij} X_{ik} + u_i \,, \; i = 1, \ldots, n \,, \tag{4.3}$$

where $s = 1, \ldots, p(p+1)/2$. It means that the squared residuals obtained from the original regression model are now regressed on squared values of the original X_i explanatory variables and their cross products. Higher powers of regressors can also be introduced [Gujarati 2004]. Note that there is a constant term in (4.3) even if the original regression does not contain it. From auxiliary regression model (4.3) new squared residuals are calculated and the coefficient of determination, R^2, is evaluated. To reject the null hypothesis that there is no heteroscedasticity, $H_0 : \alpha_1 = \alpha_2 = \ldots = \alpha_{p(p+1)/2} = 0$, the test statistics nR^2 must larger than the critical value of the chi-square distribution, χ^2_{df}, at the chosen level of significance. The nR^2 statistics obtained from the auxiliary regression model asymptotically follows the chi-square distribution with degrees of freedom equal to $p(p+1)/2$ [White 1980]. Hence, we compare

$$nR^2 \sim \chi^2_{p(p+1)/2} \,,$$

and if nR^2 does not exceed the critical chi-square value, there is no heteroscedasticity in the original regression model.

4.2 <u>Example of LWS Estimation</u>

In this section we will show one of many possible applications of White test for LWS estimation of the heteroscedastic regression model. Consider homoscedastic model (1.1) contaminated with certain amount of data generated with the same values of regression coefficients but much larger values of σ^2. White test on the resulting dataset will very likely lead to rejection of the null hypothesis indicating thus presence of heteroscedasticity. In this case the OLS estimator will be demonstrably ineffiecient [Greene 2003]. Since we do not know how much data were contaminated and what are appropriate weights we cannot use the GLS (or equivalently WLS) estimator (BLUE). The use of the LWS with high breakdown point ($c_0 = 0.5$) would solve the problem but eliminating half of available data certainly decreases the efficiency of an estimator as well. So, we would like to drop as little information as possible, yet remove all data contamination. A simple way offers White's test: we can gradually increase parameter c_0 until homoscedasticity is accepted. The smallest c_0 is then used for the LWS estimation. Let us demonstrate this strategy on the following numerical example:

- data are generated from regression model (1.1) consisting of 100 observations,
- $\beta_0 = 0,\ \beta_1 = 1$,
- 75-95% with $\mathrm{var}(e_i) = \sigma^2$ and 5-25% with $\mathrm{var}(e_i) = 100\sigma^2$ (σ =0.1).

The $\hat{\beta}_0, \hat{\beta}_1$ estimates are calculated by OLS and LWS ($c_1 = 1 - c_0^{opt}$, $c_0 = c_0^{opt}$) for 1000 randomly generated datasets. A typical dataset is given in Figure 12 along with corresponding nR^2 statistics. Since there are no outliers in our model, both estimators should converge to the correct values for $n \to \infty$. The calculated variances are

$$\mathrm{var}\left(\hat{\beta}_0^{OLS}\right) = 0.0070,\ \mathrm{var}\left(\hat{\beta}_1^{OLS}\right) = 0.0561$$

for the OLS and

$$\mathrm{var}\left(\hat{\beta}_0^{LWS}\right) = 0.0010,\ \mathrm{var}\left(\hat{\beta}_1^{LWS}\right) = 0.0053$$

for the LWS estimator. Thus, we see that in this case the LWS is indeed more efficient than OLS.

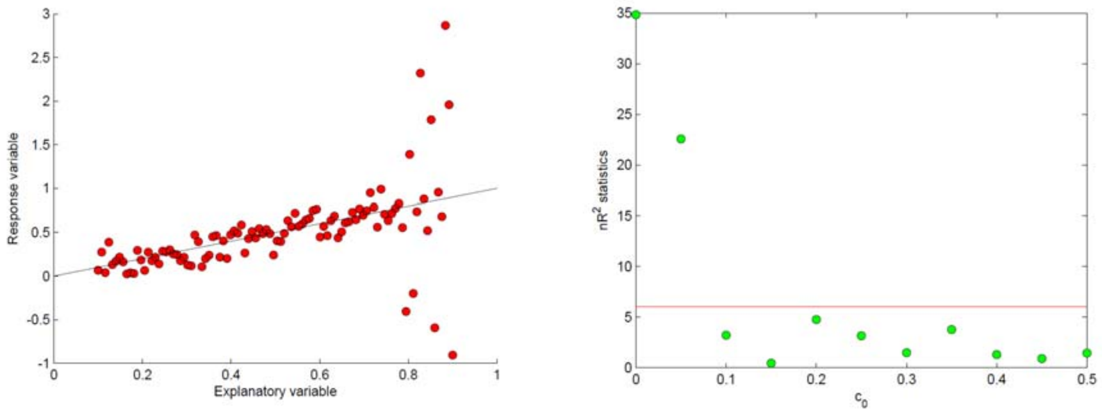

Figure 12
Dataset with 15% contamination and corresponding nR^2 statistics evaluated for $c_0 = [0, 0.5]$. The red line is the critical value of χ^2_{df}.

5. Conclusion

The main emphasis of this work is on the development and implementation of reliable and computationally efficient algorithms for solving the Least Weighted Squares problem. Though practically oriented it addresses some important issues with potentially interesting applications in the field of robust regression analysis. It has become fairly obvious that an implementation of fast and reliable algorithms is the important part of establishing any new estimator and the availability of such algorithms is the primary key to its success. From this point of view, the development of a novel exact algorithm (CP-LWS) is a significant contribution. To author's knowledge, CP-LWS is the first exact LWS algorithm running in polynomial time. It is based on the concept of geometrically feasible permutations elaborated in this work. So far CP-LWS was implemented for solving the LWS problem in two dimensional Euclidean space, a further extension to more dimensions, however, is relatively straightforward.

For many regression applications the computational efficiency is a more important criterion than exactness of results. Hence, two inexact LWS algorithms were proposed and thoroughly tested. Namely, DE-LWS based on a direct stochastic LWS optimization using Differential Evolution is superior to all currently available algoritms both in computational speed and reliability. With very modest CPU time requirements (while retaining a very high level of success probability) the newly implemented inexact algorithms are well suited for benchmarking, bootstrapping, and other computationally demanding applications.

Main results of this work can be summarized in the following points:

- usefulness of exact methods has been significantly enhanced – CP-LWS calculations on moderately large datasets (~50) are now possible.
- Fast CP-LWS method might be potentially employed for large scale calculations. At the moment, however, it relies on FastLWS with the OLS sampling. Further research on a more efficient sampling method is required to obtain comparable performance with DE-LWS.
- DE-LWS is currently the fastest and most reliable algorithm for solving the LWS optimization problem. Another advantage is the simplicity and easy implementation (even for multidimensional optimization).

In addition to many potential uses of the LWS method in robust econometric analysis (such as outlier diagnostics, etc.) we have focused, albeit very briefly, on the problem of robust estimation in the presence of heteroscedasticity. Considering how important this is for the reliability of robust regression analysis, we plan to investigate it more thoroughly in the near future. At present, it has been convincingly demonstrated that combined use of the LWS estimator and White's test for heteroscedasticity may significantly improve the efficiency of regression estimation.

All algorithms and testing programs were implemented in MATLAB and they are available for interested reader electronically on the attached CD data disc.

References

[Brest 2006] Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V. (2006): *Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems*. IEEE Trans. on Evol. Comp. 10(6), 646–657

[Greene 2003] Greene, W.H. (2003): *Econometric Analysis*. 5th Edition, New Jersey: Prentice Hall.

[Gujarati 2004] Gujarati, D. (2004): *Basic Econometrics*. 4th Edition, The McGraw-Hill Companies

[Hampel 1986] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A. (1986): *Robust Statistics: The Approach Based on Influence Functions*. Wiley series in probability and statistics.

[Plát 2004] Plát, P. (2004): *Modifikace Whiteova testu pro nejmenší vážené čtverce*. Robust 2004, 291-298.

[Price 2005] Price, K., Storn, R., Lampinen, J. (2005): *Differential Evolution – A Practical Approach to Global Optimization*. Springer, Heidelberg

[Rousseeuw 1987] Rousseeuw, P.J., Leroy, A.M. (1987): *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc.

[Rousseeuw 2002] Rousseeuw, P.J. and Van Driessen, K. (2002): *Fast-LTS in Matlab*.

[Skuhrovec 2009] Skuhrovec, J. (2009): *Analysis of LWS Empirical Properties Using Bootstrap Methods*. Diplomová práce, České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská.

[Storn 1996] Storn, R., Price, K.V. (1996): *Minimizing the real functions of the ICEC'96 contest by differential evolution*. In: Proceedings of the 1996 IEEE international conference on evolutionary computation, Nagoya, Japan, pp. 842–844. IEEE Press, New York

[Storn 1997] Storn, R., Price, K.V. (1997): *Differential Evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization 11(4), 341–359

[Víšek 1997] Víšek, J.Á. (1997): *Ekonometrie I*. Praha: Karolinum.

[Víšek 2000] Víšek, J.Á. (2000): *Regression with high breakdown point*. Robust 2000, 324-356.

[Víšek 2002a] Víšek, J.Á. (2002): *The least weighted squares I. The asymptotic linearity of* *normal equations*. Bulletin of the Czech Econometric Society, vol 9, no 15, 31-58.

[Víšek 2002b] Víšek, J.Á. (2002): *The least weighted squares II. Consistency and asymptotic normality*. Bulletin of the Czech Econometric Society, vol 9, no 16, 1-28.

[Víšek 2004] Víšek, J.Á. (2004): *The Least Weighted Squares under heteroscedasticity.* Proceedings of PRASTAN 2004, Kočovce, May 17-21,2004, eds. Kalina, M., Minárová, M., Nánásiová, O., 117 - 128.

[White 1980] White, H. (1980): *A Heteroscedasticity - Consistent Covariance Matrix Estimator and a Direct Test of Heteroscedasticity*. Econometrica, vol. 48, 817-838.

UNIVERSITAS CAROLINA PRAGENSIS
založena 1348

Univerzita Karlova v Praze
Fakulta sociálních věd
Institut ekonomických studií

Opletalova 26
110 00 Praha 1
TEL: 222 112 330,305
TEL/FAX: 222 112 304
E-mail: ies@mbox.fsv.cuni.cz
http://ies.fsv.cuni.cz

Akademický rok 2009/2010

TEZE BAKALÁŘSKÉ PRÁCE

Student:	Věra Bludská
Obor:	Ekonomie
Konzultant:	Jan Ámos Víšek

Garant studijního programu Vám dle zákona č. 111/1998 Sb. o vysokých školách a Studijního a zkušebního řádu UK v Praze určuje následující bakalářskou práci

Předpokládaný název BP:

Whitův test pro nejmenší vážené čtverce

Charakteristika tématu, současný stav poznání, případné zvláštní metody zpracování tématu:

Řešený problém spadá do oblasti robustní ekonometrie a jeho cílem je nalézt dianostický nástroj na odhalení heteroskedasticity disturbancí v regresním modelu, jehož koeficienty odhadujeme pomocí nejmenších vážených čtverců, tj. pomocí robustifikované verze běžných nejmenších čtverců. Metoda nejmenších vážených čtverců byla prostudována v řadě článku konzultanta a jeho doktorandů v situaci, kdy se předpokládá homoskedasticita disturbancí. Kvalita a spolehlivost odhadnutého regresního modelu je signifikantně závislá na předpokladu homoskedasticity disturbancí. Pokud je tento předpoklad porušen, odhadnutý model nemusí odpovídat skutečnosti, může např. zahrnovat celou řadu nesignifikantních vysvětlujících proměných, což následně může vést ke značně vychýleným odhadům regresních jkoeficientů. Proto je řešení výše popsaného problému klíčové pro spolehlivost regresní analýzy dat.

Struktura BP:

Abstrakt

Cílem práce je nalezení diagnostického nástroje pro odhalení heteroskedasticity disturbancí při robustní regresní analýze dat. Kromě teoretického odvození bude provedena i numerická studie, která pro různé typy heteroskedasticity a různé úrovně kontaminace prozkoumá chování odvozené statistiky pro konečné výběry dat.

Osnova

Zavedení formalizmu a připomenutí regresního modelu spolu s klasickou analýzou založenou na běžných nejmenších čtvercích.

Úvod do základních pojmů a cílů robustní analýzy dat.

Zavedení metody nejmenších vážených čtverců a shrnutí dosud nalezených výsledků.

Připomenutí výsledků Halberta Whita.
Nalezení robustifikované verze Whitova testu a prostudování jeho vlastností.
Numerická studie.

Seznam základních pramenů a odborné literatury:

Víšek, J. Á. (1997): *Ekonometrie I.* Nakladatelství Univerzity Karlovy v Praze.

Hampel, F. R., E. M. Ronchetti, P. J. Rousseeuw, W. A. Stahel (1986): *Robust Statistics - The Approach Based on Influence Functions.* New York: J.Wiley and Son.

White, H. (1980): A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroscedasticity. *Econometrica, 48, 817 - 838.*

Víšek, J. Á. (2000): Regression with high breakdown point. ROBUST 2000, 324 – 356, The Union of the Czech Mathematicians and Physicists and the Czech Statistical Society 2001.

Víšek, J. Á. (2002): The least weighted squares I. The asymptotic linearity of normal equations. *Bulletin of the Czech Econometric Society, Vol. 9, no. 15, 31 – 58.*

Víšek, J. Á. (2002): The least weighted squares II. Consistency and asymptotic normality. *Bulletin of the Czech Econometric Society, Vol. 9, no. 16, 1 – 28.*

Datum zadání:	1. 6. 2010
Termín odevzdání:	

Podpisy konzultanta a studenta:

V Praze dne 20. 5. 2010