Recursive Algorithm for L1 Norm Estimation in Linear Models

A. Khodabandeh[1] and A. R. Amiri-Simkooei[2]

Abstract: L1 norm estimator has been widely used as a robust parameter estimation method for outlier detection. Different algorithms have been applied for L1 norm minimization among which the linear programming problem based on the simplex method is well known. In the present contribution, in order to solve an L1 norm minimization problem in a linear model, an interior point algorithm is developed which is based on Dikin's method. The method can be considered as an appropriate alternative for the classical simplex method, which is sometimes time-consuming. The proposed method, compared with the simplex method, is thus easier for implementation and faster in performance. Furthermore, a recursive form of the Dikin's method is derived, which resembles the recursive least-squares method. Two simulated numerical examples show that the proposed algorithm gives as accurate results as the simplex method but in considerably less time. When dealing with a large number of observations, this algorithm can thus be used instead of the iteratively reweighted least-squares method and the simplex method.

DOI: 10.1061/(ASCE)SU.1943-5428.0000031

CE Database subject headings: Least squares method; Computer programming; Algorithms; Surveys; Estimation.

Author keywords: Iteratively reweighted least squares; Linear programming problem; Interior point methods; Simplex method; Affine method of Dikin; Robust estimation; L1 norm minimization; Recursive algorithm.

Introduction

The least-squares estimation, also known as best linear unbiased estimation (BLUE), is the standard and powerful method to estimate the unique and optimum solution for unknown parameters. For better statistical interpretation of the results, it is common to assume that the observation errors are random and preferably of normal distribution. When the observations are only affected by zero-mean random errors and variances and covariances of the observations are available, then the least-squares estimation of the unknown parameters is an optimal (best in the sense of minimum variance) estimation. If, however, the observations are falsified by gross errors, this optimality criterion is violated. The classical methods for outlier detection are based on statistical tests, which can be applied for normally distributed data such as data snooping (Baarda 1968) and τ-test (Pope 1976). Because the statistic of these methods is a function of adjusted residuals, the methods are efficient only when a few outliers exist in the observations (Koch 1999).

If the aforementioned assumptions are violated, e.g., in cases where the observations are affected by many gross errors in addition to random ones, the robust estimation techniques produce superior results. Such robust techniques are useful for data screening and gross error detection before a final adjustment is made by the least squares. Therefore, robust parameter estimation has to be applied to guard model against the blunders and systematic errors. Recently, many papers concerning robust estimation and outlier detection techniques have been published in geodetic literature (e.g., Yang 1994, 1999; Hekimoglu 1997; Koch 1999; Baselga 2007).

L1 norm estimation is a well-known robust estimation method. The L1 norm minimization technique uses a special sufficient (and not redundant) subset of functionally independent observations which minimizes the sum of the weighted absolute value of the other residuals. On the one hand, the technique would not result in an optimum estimation in the sense that not all of the observations have been used in this method. For example, the L1 norm gives unbiased estimates like the least squares but they are not of minimum variance. Note that when the observables have Laplace distribution (i.e., double exponential distribution), L1 norm estimator is identical to the maximum likelihood estimator (Maronna et al. 2006). On the other hand, the L1 norm estimator is less sensitive to gross errors than the least-squares estimator. For example, in the case of repeated measurements of an unknown quantity, for the purpose of the population mean estimation, the least squares yields the sample *mean*, whereas the L1 norm minimization yields the sample *median*. It is evident that a gross error will affect the sample mean more than the sample median.

An approximate method for L1 norm minimization is the iteratively reweighted least squares (IRLS), which was originally introduced by Krarup et al. (1980) into geodetic applications. The method can be applied easily, but it is based on the least-squares

[1]M.Sc. Student, Dept. of Surveying and Geomatics Engineering, Geodesy Div., Faculty of Engineering, Tehran Univ., North Kargar Ave., Amir-Abad, Tehran, Iran (corresponding author). E-mail: khodabandeh@ut.ac.ir

[2]Assistant Professor, Delft Institute of Earth Observation and Space Systems, Faculty of Aerospace Engineering, Delft Univ. of Technology, Kluyverweg 1, 2629 HS, Delft, the Netherlands; and Dept. of Surveying Engineering, Faculty of Engineering, The Univ. of Isfahan, Isfahan 81744, Iran. E-mail: a.amirisimkooei@tudelft.nl

Note. This manuscript was submitted on May 26, 2009; approved on February 24, 2010; published online on February 26, 2010. Discussion period open until July 1, 2011; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Surveying Engineering***, Vol. 137, No. 1, February 1, 2011. ©ASCE, ISSN 0733-9453/2011/1-1–8/$25.00.

principle, and hence it is not reliable when the real solution lies far away from the initial solution (Baselga 2007). And other methods like L1 norm minimization are presented in Marshall and Bethel (1996); Koch (1999); Amiri-Simkooei (2003); Junhuan (2005); and Baselga (2007). One of the efficient methods for L1 norm estimation is the simplex method (Dantzig 1963; Bazaraa et al. 1990), which performs calculations based on some pivotal rules and can thus take a large amount of computational time when the number of variables is large.

Barrodale and Roberts (1972, 1973) modified the simplex method by restricting the choice of pivotal column through certain columns for L1 norm minimization and thereby substantially reduced the number of the required iterations. In particular, if the number of the observations is not very large, the method performs quite well. However, the problem of computational loading will also arise for this method in case of large number of observations. Portnoy and Koenker (1997) demonstrated that the interior point methods improve the computational speed 10–100 times over all simplex-based methods. This holds for L1 norm minimization methods such as that presented by Barrodale and Roberts (1972, 1973) in large problems.

The lack of attention paid to the L1 norm adjustment in geodetic applications is mainly due to the relative complexity of its implementation compared to the least squares. Though this complexity should not be of great importance in the presence of modern computing systems, the present contribution proposes a new technique, which is simple in implementation.

The objectives of this paper are as follows. Since the simplex algorithm could be time-consuming, especially when the number of observations is large, we introduce interior point algorithms based on Dikin's method for solving the linear programming problems (LPPs). An iterative algorithm, with its steps involved, is presented for the L1 norm minimization in a linear model. The method is shown to be faster than the simplex method. The method will then be generalized for a recursive model (called sequential L1 norm minimization), when a new observation group is available in the functional model. Finally, the efficiency of the Dikin's method will be verified by using two numerical examples. The results of this method will be compared with those obtained from the simplex method and IRLS method.

L1 Norm Minimization in a Linear Model

Formulation as a Linear Programming Problem

L1 norm minimization is widely used as a robust technique for outlier detection in geodetic applications. The usual method for an L1 norm minimization problem leads to a linear optimization problem. The variables of a linear optimization problem are usually nonnegative, while the unknown parameters and residuals are in general real numbers. Therefore, the variables of the Gauss-Markov model should be rewritten as nonnegative variables (slack variables). Consider the following linearized model of observation equations:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{A}}\boldsymbol{\beta} + \tilde{\mathbf{e}} \tag{1}$$

where $\tilde{\mathbf{y}}$=m-vector of observables; $\tilde{\mathbf{e}}$=m-vector of random errors; $\boldsymbol{\beta}$=n-vector of unknown parameters; and $\tilde{\mathbf{A}}$=$m \times n$ design matrix of full column rank. Having uncorrelated observations, one may divide each observation $\tilde{y}_i$ by its standard deviation $\sigma_{\tilde{y}_i}$

$$y_i = \sigma_{\tilde{y}_i}^{-1}\tilde{y}_i, \quad i = 1,2,\ldots,m \tag{2}$$

to transform the original observations to those that have the same standard deviations equal to one. Therefore, by using vector notation, one can obtain an equivalent linear model $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}+\mathbf{e}$ (Koch 1999), with

$$\mathbf{y} = \mathbf{Q}_{\tilde{\mathbf{y}}}^{-1/2}\tilde{\mathbf{y}}, \quad \mathbf{e} = \mathbf{Q}_{\tilde{\mathbf{y}}}^{-1/2}\tilde{\mathbf{e}}, \quad \mathbf{A} = \mathbf{Q}_{\tilde{\mathbf{y}}}^{-1/2}\tilde{\mathbf{A}} \tag{3}$$

where the covariance matrix of the new observables $\mathbf{y}$=identity matrix, and $\mathbf{Q}_{\tilde{\mathbf{y}}}^{-1/2}$=diagonal matrix of which its diagonal elements are the entries of the vector $[\sigma_{\tilde{y}_1}^{-1}, \sigma_{\tilde{y}_2}^{-1}, \ldots, \sigma_{\tilde{y}_m}^{-1}]^T$. Therefore, for the sake of simplicity and notational convenience, in the sequel, we work with the transformed linear model $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}+\mathbf{e}$.

Using the fact that any real number can be written as a difference of two nonnegative real numbers, one may write (Marshall and Bethel 1996; Amiri-Simkooei 2003)

$$\sum_{i=1}^{m} |e_i| = \sum_{i=1}^{m} |u_i - v_i| = \sum_{i=1}^{m} (u_i + v_i) \rightarrow \min \tag{4}$$

where each of the entries of u and v should fulfill either ($u_i \geq 0$ and $v_i=0$) or ($v_i \geq 0$ and $u_i=0$). In vector form, one may write

$$\mathbf{e} = \mathbf{u}\text{-}\mathbf{v}, \quad \boldsymbol{\beta} = \boldsymbol{\lambda}\text{-}\boldsymbol{\gamma} \tag{5}$$

where the nonnegative vectors $\mathbf{u}$ and $\mathbf{v}$ are of size m, and the nonnegative vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ are of size n. The linear model $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}+\mathbf{e}$ with an identity covariance matrix for the observable $\mathbf{y}$, along with the L1 norm criterion in Eq. (4) can be reformulated to a standard linear programming as (Amiri-Simkooei 2003)

$$\text{Minimize} \quad z = \mathbf{f}^T\mathbf{x} = [\mathbf{0}^T \vdots \mathbf{0}^T \vdots \mathbf{c}_m^T \vdots \mathbf{c}_m^T] \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix} \tag{6a}$$

$$\text{Subject to} \quad \mathbf{B}^T\mathbf{x} = [\mathbf{A} \vdots -\mathbf{A} \vdots \mathbf{I}_m \vdots -\mathbf{I}_m] \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{y} \tag{6b}$$

where c_m=m-vector containing all ones (a summation vector); $\mathbf{0}$=n-vector containing all zeros; and I_m=identity matrix of size m. The objective function in Eq. (6*a*) with constraint in Eq. (6*b*) can be solved either by the simplex method (Dantzig 1963) or by the interior point method (Roos et al. 2005). We now introduce the interior point methods based on Dikin's algorithm which is considered to be faster in computational speed and easier for implementation than the simplex method.

Affine Method of Dikin

The usual method for solving a LPP is the classical simplex method developed by Dantzig (1963), which is recognized to be efficient in implementation but inefficient in speed. The alternative methods are the interior point algorithms. They improve a feasible interior solution point of the linear programming by steps through the interior, rather than a point that improves stepwise around the boundary of the feasible region like that in the simplex method.

One of the simplest methods of such algorithms is Dikin's method (Dantzig and Thapa 1997), which can easily be applied. In this method, an initial interior feasible point is given, or it can

be found. The method then improves the interior feasible point in an iterative procedure. To see this, consider the following LPP in the standard form:

$$\text{Minimize} \quad z = \mathbf{f}^T\mathbf{x}, \quad \text{Subject to} \quad \mathbf{B}^T\mathbf{x} = \mathbf{b}; \mathbf{x} \geq \mathbf{0} \tag{7}$$

where z=objective function; $\mathbf{f}$ and $\mathbf{x}$=vectors of identical size in which the entries of $\mathbf{f}$ are known (given numbers) and the entries of $\mathbf{x}$ are some unknown nonnegative variables to be determined; and $\mathbf{B}$ and $\mathbf{b}$=constraint matrix and constraint vector, respectively. In order to improve the current feasible solution $\mathbf{x}^{(t)}$ (superscript t refers to the iteration counter), one can move $\mathbf{x}^{(t)}$ in the negative gradient of the objective function, i.e., steepest descent of the objective function. The new solution $\mathbf{x}^{(t+1)}$ should also be feasible, which implies that

$$\mathbf{B}^T\mathbf{x}^{(t+1)} = \mathbf{B}^T(\mathbf{x}^{(t)} - \alpha\mathbf{f}) = \mathbf{b} \tag{8}$$

where α=positive scalar. Eq. (8) can be simplified as

$$\mathbf{B}^T\mathbf{f} = \mathbf{0} \therefore \mathbf{f} \in N(\mathbf{B}^T) \tag{9}$$

where $N(\mathbf{B}^T)$=null space of the matrix $\mathbf{B}^T$. But, the coefficient vector $\mathbf{f}$, in general, does not belong to $N(\mathbf{B}^T)$. To overcome this drawback, one can orthogonally project the vector $\mathbf{f}$ onto the $N(\mathbf{B}^T)$ as follows:

$$\mathbf{d} = -\mathbf{P}_B^{\perp}\mathbf{f} \tag{10}$$

where $\mathbf{d}$=direction vector and the projection matrix (orthogonal projector) $\mathbf{P}_B^{\perp}$ is given as

$$\mathbf{P}_B^{\perp} = \mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T \tag{11}$$

with $\mathbf{I}$=identity matrix.

When $\mathbf{x}^{(t)}$ is close to the boundary of the feasible region, little improvement will be given to the solution in each iteration (Dantzig and Thapa 1997). This makes the method to become too slow. The idea of the central path theory, which is one of the most important foundations for the development of interior point algorithms, has been proposed to make the algorithm fast (see Roos et al. 2005). An interior point $\mathbf{x}^{(t)}$ should thus be always in the center of the feasible region, which is achieved using the following scaling transformation:

$$\breve{\mathbf{x}} = \mathbf{D}^{-1}\mathbf{x}, \quad \breve{\mathbf{B}} = \mathbf{D}\mathbf{B}, \quad \breve{\mathbf{f}} = \mathbf{D}\mathbf{f} \tag{12}$$

where $\breve{\mathbf{x}}$, $\breve{\mathbf{B}}$, and $\breve{\mathbf{f}}$=transformed quantities, and $\mathbf{D}$=diagonal matrix of which the diagonal elements are the entries of the vector $\mathbf{x}$, namely

$$\mathbf{D} = \text{diag}(\mathbf{x}) \tag{13}$$

where diag=diagonal operator. Therefore, an improved feasible solution $\mathbf{x}^{(t+1)}$ can then be obtained as

$$\breve{\mathbf{x}}^{(t+1)} = \breve{\mathbf{x}}^{(t)} + \alpha\mathbf{d} \tag{14}$$

The step-size parameter α is strictly set to be between zero and $1/\theta$ (Roos et al. 2005), where the scalar θ=minimum element of the direction vector $\mathbf{d}$ with negative sign

$$\theta = -\min \mathbf{d} \tag{15}$$

For most practical applications, one may choose the scalar α as $\alpha = 0.9/\theta$ (Dantzig and Thapa 1997). This procedure should be iterated until $\mathbf{x}^{(t+1)}$ becomes close enough to $\mathbf{x}^{(t)}$ such that further iterations do not improve $\mathbf{x}^{(t)}$. For more explanation, we refer to Dantzig and Thapa (1997) and Roos et al. (2005).

In the above procedure, an initial feasible interior point solution should be available as a starting point. One may use the "Big-M" method (Dantzig and Thapa 1997) to generate this point. Therefore, the LPP of Eq. (7) may change to

$$\text{Minimize} \quad z = \breve{\mathbf{f}}^T\breve{\mathbf{x}} + Mx_\alpha \tag{16a}$$

$$\text{Subject to} \quad \breve{\mathbf{B}}^T\breve{\mathbf{x}} + (\mathbf{b} - \breve{\mathbf{B}}^T\breve{\mathbf{x}}_o)x_\alpha = \mathbf{b} \tag{16b}$$

where M=large scalar constant (e.g., $M = 1,000$) and the scalar x_α=artificial variable. When the new artificial variable x_α is added, the problem is solved with the new objective function. It should be noted that the corresponding constraint matrix will also change. Provided the coefficient M is chosen large enough, it is clear that x_α will be driven to zero when minimizing the objective function. By this trick, an arbitrary solution point $\breve{\mathbf{x}} = \breve{\mathbf{x}}_o$ and $x_\alpha = 1$ satisfies the feasibility condition. The steps of Dikin's algorithm to solve for the L1 norm minimization in a linear model are summarized in the Appendix (Algorithm 1). The solution is achieved in an iterative procedure.

Recursive Algorithm

Suppose that two successive systems of observation equations are given. The observable data of the first system are collected in the vector $\mathbf{y}_{[k-1]} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_{k-1}^T]^T$, which consists of $k-1$ groups of observations. Each group of observations $\mathbf{y}_i$, $i = 1, \ldots, k-1$, is related to the unknown parameters $\boldsymbol{\beta}$ of the functional model as $\mathbf{y}_i = \mathbf{A}_i\boldsymbol{\beta} + \mathbf{e}_i$, where $\mathbf{A}_i$ and $\mathbf{e}_i$ are their corresponding design matrix and residuals, respectively. For the observable vector $\mathbf{y}_{[k-1]}$, the corresponding unknown parameters $\mathbf{x}_{[k-1]}$ need to be determined by a recursive algorithm ($k \geq 2$). Now the observable vector $\mathbf{y}_k$ is added at step k, which makes the second system of observation equations, $\mathbf{y}_k = \mathbf{A}_k\boldsymbol{\beta} + \mathbf{e}_k$.

We now compute the updated values for $\mathbf{x}_{[k]}$ (of the observables $\mathbf{y}_{[k]} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T]^T = [\mathbf{y}_{[k-1]}^T, \mathbf{y}_k^T]^T$) and determine the new variables $\mathbf{u}_k$ and $\mathbf{v}_k$. For this purpose, consider the following LPP of the first system ($k \geq 2$):

$$\text{Minimize} \quad \mathbf{f}_{[k-1]}^T\mathbf{x}_{[k-1]} = \begin{bmatrix}\mathbf{0}^T & \mathbf{0}^T & \mathbf{c}_m^T & \mathbf{c}_m^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \mathbf{u}_{(k-1)} \\ \mathbf{v}_{(k-1)}\end{bmatrix} \tag{17a}$$

$$\text{Subject to} \quad \mathbf{B}_{[k-1]}^T\mathbf{x}_{[k-1]} = \begin{bmatrix}\mathbf{A}_{(k-1)} & -\mathbf{A}_{(k-1)} & \mathbf{I}_m & -\mathbf{I}_m\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda} \\ \boldsymbol{\gamma} \\ \mathbf{u}_{(k-1)} \\ \mathbf{v}_{(k-1)}\end{bmatrix} = \mathbf{y}_{[k-1]} \tag{17b}$$

where $\mathbf{u}_{[k-1]} = [\mathbf{u}_1^T, \ldots, \mathbf{u}_{k-1}^T]^T$, $\mathbf{v}_{[k-1]} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_{k-1}^T]^T$, and $\mathbf{A}_{[k-1]} = [\mathbf{A}_1^T, \ldots, \mathbf{A}_{k-1}^T]^T$=design matrix corresponding to the observable vector $\mathbf{y}_{[k-1]}$. The total number of observables, up to step $k-1 = m = \sum_{i=1}^{k-1} m_i$, where the number of the observations at step $i = m_i$ ($i = 1, 2, \ldots, k-1$). When the new observable vector $\mathbf{y}_k$ (of size m_k) is added, the LLP modifies to

$$\text{Minimize } \mathbf{f}_{[k]}^T\mathbf{x}_{[k]} = \begin{bmatrix}\mathbf{0}^T & \mathbf{0}^T & \mathbf{c}_m^T & \mathbf{c}_m^T & \mathbf{c}_{m_k}^T & \mathbf{c}_{m_k}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\gamma}\\ \mathbf{u}_{(k-1)}\\ \mathbf{v}_{(k-1)}\\ \mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix} \tag{18a}$$

$$\text{Subject to } \mathbf{B}_{[k]}^T\mathbf{x}_{[k]} = \begin{bmatrix}\mathbf{A}_{(k-1)} & -\mathbf{A}_{(k-1)} & \mathbf{I}_m & -\mathbf{I}_m & \mathbf{0} & \mathbf{0}\\ \mathbf{A}_k & -\mathbf{A}_k & \mathbf{0} & \mathbf{0} & \mathbf{I}_{m_k} & -\mathbf{I}_{m_k}\end{bmatrix} \times \begin{bmatrix}\boldsymbol{\lambda}\\ \boldsymbol{\gamma}\\ \mathbf{u}_{(k-1)}\\ \mathbf{v}_{(k-1)}\\ \mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix} = \begin{bmatrix}\mathbf{y}_{(k-1)}\\ \mathbf{y}_k\end{bmatrix} \tag{18b}$$

Having $\mathbf{x}_{[k-1]}$, $\mathbf{y}_k$, and $\mathbf{A}_k$, the goal now is to obtain $\mathbf{x}_{[k]}$ without storing the previous data $\mathbf{y}_{[k-1]}$. According to Eq. (13), the diagonal matrix $\mathbf{D}_{[k]}$ is obtained as

$$\mathbf{D}_{[k]} = \begin{bmatrix}\mathbf{D}_{[k-1]} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_k\end{bmatrix} \tag{19}$$

where

$$\mathbf{D}_{[k-1]} = \text{diag}(\mathbf{x}_{[k-1]}) \quad \text{and} \quad \mathbf{D}_k = \text{diag}\left(\begin{bmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix}\right) \tag{20}$$

Therefore, the transformed quantities are obtained by Eq. (12) as follows:

$$\breve{\mathbf{B}}_{[k]} = \mathbf{D}_{[k]}\mathbf{B}_{[k]} = \begin{bmatrix}\mathbf{D}_{[k-1]}\mathbf{B}_{[k-1]} & \mathbf{D}_{[k-1]}\mathbf{H}_k\\ \mathbf{0} & \mathbf{D}_k\mathbf{J}_k\end{bmatrix}$$

and

$$\breve{\mathbf{f}}_{[k]} = \mathbf{D}_{[k]}\mathbf{f}_{[k]} = \begin{bmatrix}\mathbf{D}_{(k-1)}\mathbf{f}_{(k-1)}\\ \mathbf{D}_k\mathbf{f}_k\end{bmatrix} \tag{21}$$

where $\mathbf{f}_k = [\mathbf{c}_{m_k}^T \mathbf{c}_{m_k}^T]^T$ and

$$\mathbf{H}_k = \begin{bmatrix}\mathbf{A}_k^T\\ -\mathbf{A}_k^T\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \quad \text{and} \quad \mathbf{J}_k = \begin{bmatrix}\mathbf{I}_{m_k}\\ -\mathbf{I}_{m_k}\end{bmatrix} \tag{22}$$

The transformed constraint matrix $\breve{\mathbf{B}}_{[k]}$ consists of two submatrices $\mathbf{E}$ and $\mathbf{F}$ as

$$\breve{\mathbf{B}}_{[k]} = [\mathbf{E} \vdots \mathbf{F}] \tag{23}$$

where $\mathbf{E}$ consists of the transformed constraint matrix $\breve{\mathbf{B}}_{[k-1]}$ and $\mathbf{F}$ is obtained by the new observations $\mathbf{y}_k$. One can now apply an orthogonal decomposition to $\breve{\mathbf{B}}_{[k]} = [\mathbf{E} \vdots \mathbf{F}]$ (Teunissen et al. 2005) from which the new matrix $\bar{\mathbf{F}}$ is obtained. Columns of $\bar{\mathbf{F}}$ are orthogonal to the columns of $\mathbf{E}$ and the column space of $[\mathbf{E} \vdots \bar{\mathbf{F}}]$ is equal to the column space of $[\mathbf{E} \vdots \mathbf{F}]$. That is

$$\bar{\mathbf{F}} = \mathbf{P}_{\mathrm{E}}^{\perp}\mathbf{F}, \quad \mathbf{P}_{\mathrm{E}}^{\perp} = \mathbf{I} - \mathbf{E}(\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T \tag{24}$$

One can simply show that

$$\mathbf{P}_{\mathrm{E}}^{\perp} = \begin{bmatrix}\mathbf{P}_{\mathrm{B}_{[k-1]}}^{\perp} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix} \tag{25a}$$

where

$$\mathbf{P}_{\mathrm{B}_{[k-1]}}^{\perp} = \mathbf{I} - \mathbf{D}_{[k-1]}\mathbf{B}_{[k-1]}(\mathbf{B}_{[k-1]}^T\mathbf{D}_{[k-1]}^2\mathbf{B}_{[k-1]})^{-1}\mathbf{B}_{[k-1]}^T\mathbf{D}_{[k-1]} \tag{25b}$$

Therefore, the projector matrix $\mathbf{P}_{\mathrm{B}_{[k]}}^{\perp}$ is obtained as follows:

$$\mathbf{P}_{\mathrm{B}_{[k]}}^{\perp} = \mathbf{P}_{\mathrm{E}}^{\perp} - \mathbf{P}_{\bar{\mathbf{F}}} \tag{26a}$$

where

$$\mathbf{P}_{\bar{\mathbf{F}}} = \mathbf{D}_k\bar{\mathbf{F}}(\bar{\mathbf{F}}^T\mathbf{D}_k^2\bar{\mathbf{F}})^{-1}\bar{\mathbf{F}}^T\mathbf{D}_k \tag{26b}$$

Similar to the decomposition of Eq. (23), the direction vector consists of two parts

$$\mathbf{d} = \begin{bmatrix}\Delta\mathbf{x}_{(k-1)}\\ \Delta\mathbf{w}_k\end{bmatrix} \tag{27}$$

where the vector $\Delta\mathbf{x}_{[k-1]}$ updates the variables vector $\mathbf{x}_{[k-1]}$ and $\Delta\mathbf{w}_k$ computes the added variables $\mathbf{u}_k$ and $\mathbf{v}_k$. Substituting the above equations into Eq. (10), after a few simple matrix operations, yields

$$\Delta\mathbf{x}_{[k-1]} = (\mathbf{Q}_{11} - \mathbf{P}_{B_{k-1}}^{\perp})\mathbf{D}_{[k-1]}\mathbf{f}_{[k-1]} + \mathbf{Q}_{12}\mathbf{D}_k\mathbf{f}_k \tag{28a}$$

$$\Delta\mathbf{w}_k = (\mathbf{Q}_{22} - \mathbf{I})\mathbf{D}_k\mathbf{f}_k + \mathbf{Q}_{12}^T\mathbf{D}_{[k-1]}\mathbf{f}_{[k-1]} \tag{28b}$$

where

$$\mathbf{Q}_{11} = \mathbf{P}_{\mathrm{B}_{k-1}}^{\perp}\mathbf{D}_{[k-1]}\mathbf{H}_k(\bar{\mathbf{F}}^T\bar{\mathbf{F}})^{-1}\mathbf{H}_k^T\mathbf{D}_{[k-1]}\mathbf{P}_{\mathrm{B}_{k-1}}^{\perp} \tag{29a}$$

$$\mathbf{Q}_{12} = \mathbf{P}_{\mathrm{B}_{k-1}}^{\perp}\mathbf{D}_{[k-1]}\mathbf{H}_k(\bar{\mathbf{F}}^T\bar{\mathbf{F}})^{-1}\mathbf{J}_k^T\mathbf{D}_k \tag{29b}$$

$$\mathbf{Q}_{22} = \mathbf{D}_k\mathbf{J}_k(\bar{\mathbf{F}}^T\bar{\mathbf{F}})^{-1}\mathbf{J}_k^T\mathbf{D}_k \tag{29c}$$

Furthermore, according to the Big-M method and Eqs. (16) and (17), the second part of Eq. (18*b*) will change as

$$\mathbf{H}_k^T\mathbf{x}_{[k-1]} + \mathbf{J}_k^T\begin{bmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix} + \left[\mathbf{y}_k - \mathbf{H}_k^T\mathbf{x}_{(k-1)}^o - \mathbf{J}_k^T\begin{pmatrix}\mathbf{u}_k^o\\ \mathbf{v}_k^o\end{pmatrix}\right]x_\alpha = \mathbf{y}_k \tag{30}$$

where the initial point solution $\mathbf{x}_{[k-1]}^o$ must be set to the computed variables at step k-1 and the added variables $(\mathbf{u}_k^o, \mathbf{v}_k^o)$ can take arbitrary values. By this trick, all of the observable data $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{k-1})$ are not necessary to be available at step k, which is the main advantage of the recursive algorithm. This fact prompts the algorithm to compute the variables rapidly.

Finally, by using Eq. (14), the transformed updated variables $\breve{\mathbf{x}}_{[k-1]}$ and the transformed added variables $\breve{\mathbf{u}}_k$ and $\breve{\mathbf{v}}_k$ are obtained by an iterative procedure

$$\breve{\mathbf{x}}_{[k-1]}^{(t+1)} = \breve{\mathbf{x}}_{[k-1]}^{(t)} + \alpha\Delta\mathbf{x}_{[k-1]} \quad \text{(updating equation)} \tag{31a}$$

$$\begin{bmatrix}\breve{\mathbf{u}}_k\\ \breve{\mathbf{v}}_k\end{bmatrix}^{(t+1)} = \begin{bmatrix}\breve{\mathbf{u}}_k\\ \breve{\mathbf{v}}_k\end{bmatrix}^{(t)} + \alpha\Delta\mathbf{w}_k \tag{31b}$$

where the scalar α is set to $\alpha = 0.9/\theta$ and the scalar θ is given by Eq. (15). This procedure can be applied sequentially. In each step,

Table 1. Observations and Adjusted Residuals of the Leveling Network in Example 1

Observation number	From	To	Δh(m)	Residuals (mm) L1 (simplex)	L1 (Dikin)	L1 (IRLS)	Least squares
1	P1	P2	1.9888	5.17	5.17	8.40	12.23
2	P2	P3	1.5062	61.22	61.22	55.09	39.05
3	P3	P4	1.5827	2.68	2.68	5.70	23.94
4	P4	P5	−5.8090	0.00	0.00	−0.13	−2.24
5	P5	P1	0.8014	1.02	1.02	0.03	−2.88
6	P1	P3	3.4286	0.00	0.00	−1.90	−15.11
7	P2	P5	−2.7815	2.50	2.50	0.27	−0.64
8	P2	P4	3.0172	−21.27	−21.27	−28.07	−26.18
Absolute sum:				93.86	93.86	99.59	122.29

the nonnegative variables of previous step $\mathbf{x}_{[k-1]}$ and the projector matrix $\mathbf{P}^{\perp}_{\mathrm{B}_{k-1}}$ with new observable vector $\mathbf{y}_k$ and the corresponding design matrix $\mathbf{A}_k$ should be available, but the previous observable vector $\mathbf{y}_{[k-1]}=[\mathbf{y}_1^T,\ldots,\mathbf{y}_{k-1}^T]^T$ doesn't need to be saved anymore. This algorithm is closely similar to recursive least-squares method, although they are different in principle. The structure of this algorithm is given in the Appendix (Algorithm 2).

Since the above derivations rely on the transformed linear model $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}+\mathbf{e}$, the Dikin's method and the recursive algorithm give the computed nonnegative variables corresponding to the transformed residuals $\mathbf{e}$. To achieve the adjusted residuals $\tilde{\mathbf{e}}$ corresponding to the original linear model Eq. (1), one can easily perform the inverse of the transformation introduced in Eq. (3) (i.e., $\tilde{\mathbf{e}}=\mathbf{Q}_{\tilde{\mathbf{y}}}^{1/2}\mathbf{e}$), where $\mathbf{Q}_{\tilde{\mathbf{y}}}^{1/2}$=diagonal matrix of which its diagonal elements are the standard deviations of the original observable vector $\tilde{\mathbf{y}}$.

Numerical Results and Discussions

In order to illustrate the efficiency of the proposed method, two numerical examples are presented. The first example is a leveling network (as a linear model), while the second example is a horizontal geodetic network (as a nonlinear model).

Example 1 (Linear Model)

In the first example, a simulated leveling network is assumed. Five Network Points P1, P2, P3, P4, and P5 construct the network with eight height differences observed between the points. The height of the Station P1, providing the datum of the network, is assumed to be known. Therefore, the degree of freedom of the network is df=4. The observations are assumed to have the same standard deviation of $\sigma=10$ mm. It is also assumed that a blunder with a magnitude of 70 mm occurs in the second observation (i.e., height difference P2 to P3).

Table 1 shows the observations (m) as well as the adjusted residuals (mm) for each of the methods. It can be seen that the results of the simplex and Dikin methods are the same; they are both more sensitive to outliers than the other methods. Furthermore, the lack of robustness of the least-squares estimator is illustrated when observations are contaminated with gross errors.

In the following example, the nonlinear model is considered.

Example 2 (Nonlinear Model)

In the second example, a simulated trilateration network is assumed. The network consists of seven points with 15 distance observations for the first step. The datum of the network is provided by two Control Points P1 and P2, and the degree of freedom of the network is df=5. The observations are assumed to have no gross errors except for the first observation which is erroneous by 100 mm. In the second step, two other distance observations are added, which are assumed to have no gross errors and finally, in the third step, three distance observations are added. Therefore, the degree of freedom of the network will increase to df=10. The observation number 19 (from step 3) is erroneous by 60 mm. The same standard deviation of $\sigma=10$ mm is assumed for all observations. Fig. 1 illustrates the position of the points with the distance observations. The observed distances along with their adjusted residuals for different situations are illustrated in Table 2.

The residuals estimated by both the simplex method and the recursive algorithm based on Dikin's method are more realistic compared with other adjusted residuals. Furthermore, the recursive algorithm can estimate the residuals sequentially.

The results presented in Table 2 show that the estimated residuals using Dikin's method are different from those given by simplex method for a few observations. Note however that both methods have an identical absolute residuals sum. This is what is probable to occur in practice because the L1 norm estimator is not

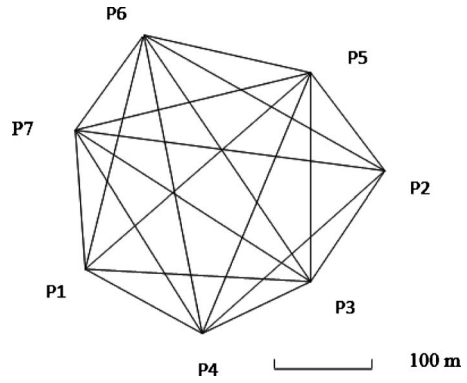

Fig. 1. Configuration and distance observations of network in Example 2

Table 2. Observations and Adjusted Residuals of the Network in Example 2

Observation number	Distances From	Distances To	Observed values (m)	Residuals (recursive) (mm) Step (1)	Step (2)	Step (3)	Residuals of the other methods (mm) L1 (Dikin)	L1 (simplex)	L1 (IRLS)	Least squares
1	1	3	203.237	81.0	81.3	82.4	82.6	82.6	70.5	52.1
2	1	4	120.083	0.0	0.0	−4.0	−4.4	−4.3	−15.3	−17.8
3	1	5	274.915	0.0	0.0	4.5	4.7	4.7	1.2	−9.1
4	1	6	229.005	0.0	0.0	0.0	0.0	0.0	0.9	7.2
5	1	7	129.153	0.0	0.0	−0.8	−0.7	−0.8	−1.2	8.6
6	2	3	147.634	15.0	15.0	15.7	15.6	15.7	21.8	26.1
7	2	4	244.333	2.5	2.7	0.0	0.0	0.0	2.9	8.1
8	2	5	95.359	0.0	0.0	0.0	0.0	0.0	7.3	13.1
9	2	6	240.578	0.0	0.0	0.1	0.0	0.0	3.9	10.7
10	2	7	283.081	0.0	0.0	−2.3	−2.2	−2.3	−6.2	−6.7
11	3	4	108.314	0.0	0.0	0.0	0.0	0.0	−4.9	−13.6
12	3	5	193.800	−8.8	−8.9	−4.0	−4.1	−4.0	−2.6	−9.1
13	3	6	273.941	−2.3	−2.3	0.0	0.0	0.0	−3.2	−8.8
14	3	7	252.880	0.0	0.0	0.0	0.0	0.0	−13.7	−27.0
15	4	5	260.669	0.0	0.0	−0.1	−0.1	0.0	−3.6	−4.0
16	4	6	283.370		5.3	0.0	0.0	0.0	−2.0	5.3
17	4	7	218.337		10.1	13.1	12.6	12.7	1.0	9.1
18	5	6	152.313			−2.9	−2.6	−2.5	−7.6	−10.5
19	5	7	219.884			53.7	53.8	53.8	41.8	29.1
20	6	7	113.619			0.0	0.0	0.0	−2.6	−7.3
Absolute sum:						183.6	183.4	183.4	214.2	283.3

guarantee to be unique. Moreover, the recursive algorithm does not deliver the absolute residuals sum that is obtained by simplex and Dikin's methods. To interpret this discrepancy, one can argue that, similar to any sequential algorithm, the inaccuracy caused by the limitation of the effective digits of the computer might be accumulated and propagated in further computing steps. This problem also occurs in the recursive least-squares processing in which the drift increases with increasing the steps (Xu 2007). Another possible cause of this discrepancy is due to the nonlinear behavior of the model which in turn depends on the approximate parameters. In other words, the recursive algorithm applies the corrections to the parameters only once rather than iteration on the parameters until the corrections become zeros.

In this example, the analysis is applied and executed to 100 simulated data sets using both the simplex method and recursive algorithm. The goal is to evaluate the performance of these two methods. The computational time of the methods is illustrated in Fig. 2. It can be seen that the recursive algorithm is approximately four times faster than the simplex method in this example. The superiority of the Dikin's method over simplex method becomes more distinguished when the number of observations in the network increases. The proposed algorithm can thus be applied when the number of the observations is large.

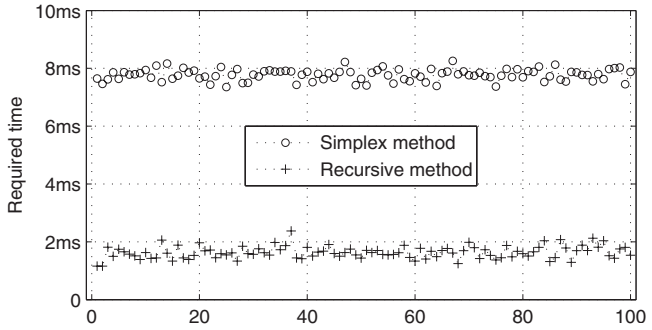

Fig. 2. Required time to compute the results of L1 norm minimization in Example 2 at 100 trials using recursive method (+) and simplex method (o)

Conclusions

When the observations are affected by gross errors, one of the well-known methods for outlier detection is the L1 norm estimation. The IRLS method, which is an approximate method for L1 norm minimization, does not seem to be always reliable. Therefore, other efficient methods such as those based on the linear programming methods need to be used. In practice, the interior point algorithms have been shown to deliver results which are consistent with the classical simplex method. The method is considered to be much faster than the simplex method. In this paper, the L1 norm minimization problem is solved sequentially by the affine method of Dikin. Furthermore, by applying the Big-M method, this algorithm does not need the previous observations at the latest step and can be applied in a stepwise manner. Similar to the recursive least-squares method, the recursive L1 norm estimation was developed, which was aimed to perform the calculations in a considerably shorter amount of time. Two numerical examples illustrated that the proposed methodology is feasible.

Acknowledgments

We would like to acknowledge the editor-in-chief and three anonymous reviewers for their useful comments, which significantly improved the presentation of this paper.

Appendix. Symbolic Representation of the Algorithms

Algorithm 1: an iterative algorithm to solve an L1 norm minimization problem for linear model $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}+\mathbf{e}$ using Dikin's algorithm

Input:

- m-vector of observations $\mathbf{y}$;
- $m\times n$ design matrix $\mathbf{A}$; and
- Small value for ε.

Begin

o Obtain $\mathbf{b}$, $\mathbf{f}$, and $\mathbf{B}$ from Eq. (6);

o Set iteration counter: $0\rightarrow t$; and

o Initialize a feasible point $\mathbf{x}^{(0)}$ satisfying $\mathbf{B}^T\mathbf{x}=\mathbf{b}$.

Begin Loop

• Create transformation matrix $\mathbf{D}=\text{diag}(\mathbf{x}^{(t)})$;

• Compute $\check{\mathbf{B}}$, $\check{\mathbf{f}}$ and, $\check{\mathbf{x}}^{(t)}$ using Eq. (12);

• Create projection matrix $\mathbf{P}_{\check{B}}^{\perp}=\mathbf{I}-\mathbf{DB}(\mathbf{B}^{\mathrm{T}}\mathbf{D}^2\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{D}$;

• Compute direction vector $\mathbf{d}=-\mathbf{P}_{\check{B}}^{\perp}\check{\mathbf{f}}$ and set $\theta=-\min\mathbf{d}$;

• Set $\alpha=0.9/\theta$ and compute $\check{\mathbf{x}}^{(t+1)}=\check{\mathbf{x}}^{(t)}+\alpha\mathbf{d}$;

• Transform back to original space:$\mathbf{x}^{(t+1)}=\mathbf{D}\check{\mathbf{x}}^{(t+1)}$;

• **While** $\|\mathbf{x}^{(t+1)}-\mathbf{x}^{(t)}\|>\varepsilon$ **repeat**; and

• Increase t by one step: $t+1\rightarrow t$.

End Loop

End

Output:

- Nonnegative optimal solution $\mathbf{x}$, and hence
- n-vector $\boldsymbol{\beta}$ of unknowns and m-vector $\mathbf{e}$ of residuals.

Algorithm 2: recursive algorithm to solve an L1 norm minimization problem for linear model $\mathbf{y}_{[k]}=\mathbf{A}_{[k]}\boldsymbol{\beta}+\mathbf{e}_{[k]}$

Input:

- The nonnegative variables of previous step $\mathbf{x}_{[k-1]}$ with $\mathbf{P}_{\mathrm{B}_{[k-1]}}^{\perp}$;
- m_k- vector of new observable $\mathbf{y}_k$ with $m_k\times n$ design matrix $\mathbf{A}_k$; and
- Small value for ε.

Begin

o Set iteration counter: $0\rightarrow t$;and

o Initialize a feasible point $\mathbf{x}^{(0)}$ using Eq. (30).

Begin Loop

• Create transformation matrices $\mathbf{D}_{[k-1]}$ and $\mathbf{D}_k$ using Eq. (20);

• Create matrices $\mathbf{H}_k$ and $\mathbf{J}_k$ using Eq. (22);

• Compute $\mathbf{Q}_{11}$, $\mathbf{Q}_{12}$, and $\mathbf{Q}_{22}$ using Eq. (29);

• Compute $\Delta\mathbf{x}_{[k-1]}$ and $\Delta\mathbf{w}_k$ using Eq. (28);

• Set $\theta=-\min\left\{\begin{bmatrix}\Delta x_{(k-1)}\\ \Delta\mathbf{w_k}\end{bmatrix}\right\}$ and $\alpha=0.9/\theta$;

• Scaling transformation: compute $\check{\mathbf{x}}_{[k-1]}^{(t)}=\mathbf{D}_{[k-1]}^{-1}\mathbf{x}_{[k-1]}^{(t)}$ and $\begin{bmatrix}\check{\mathbf{u}}_k\\ \check{\mathbf{v}}_k\end{bmatrix}^{(t)}=\mathbf{D}_k^{-1}\begin{bmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix}^{(t)}$;

• Compute $\check{\mathbf{x}}_{[k-1]}^{(t+1)}$ and $\begin{bmatrix}\check{\mathbf{u}}_k\\ \check{\mathbf{v}}_k\end{bmatrix}^{(t+1)}$ by Eq. (31);

• Transform back to original space by $\mathbf{x}_{[k-1]}^{(t+1)}=\mathbf{D}_{[k-1]}\check{\mathbf{x}}_{[k-1]}^{(t+1)}$ and $\begin{bmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix}^{(t+1)}=\mathbf{D}_k\begin{bmatrix}\check{\mathbf{u}}_k\\ \check{\mathbf{v}}_k\end{bmatrix}^{(t+1)}$;

• **While** $\|\mathbf{x}^{(t+1)}-\mathbf{x}^{(t)}\|>\varepsilon$ **repeat**; and

• Increase t by one step: $t+1\rightarrow t$.

End Loop

o Set $[\mathbf{x}_{[k-1]}^T,\mathbf{u}_k^T,\mathbf{v}_k^T]^T\rightarrow\mathbf{x}_{[k]}$ and create $\mathbf{P}_{\mathrm{B}_{[k]}}^{\perp}$ by Eq. (26).

End

Output:

- The nonnegative variables $\mathbf{x}_{[k]}$ with $\mathbf{P}_{\mathrm{B}_{[k]}}^{\perp}$.

Notation

The following symbols are used in this paper:

$\tilde{\mathbf{A}}$ = design matrix corresponding to original observable vector $\tilde{\mathbf{y}}$;

$\mathbf{A}_k$ = design matrix corresponding to observable vector $\mathbf{y}_k$;

$\mathbf{A}_{[k-1]}=[\mathbf{A}_1^T,\ldots,\mathbf{A}_{k-1}^T]^T$ = design matrix corresponding to observable vector $\mathbf{y}_{[k-1]}=[\mathbf{y}_1^T,\ldots,\mathbf{y}_{k-1}^T]^T$;

$\mathbf{D}_k$ = diagonal matrix of which its diagonal elements are entries of the vector $\begin{bmatrix}\mathbf{u}_k\\ \mathbf{v}_k\end{bmatrix}$;

$\mathbf{D}_{[k-1]}$ = diagonal matrix of which its diagonal elements are entries of vector $\mathbf{x}_{[k-1]}$;

m_k = number of observations at step k;

$N(.)$ = null space of matrix;

P_B = orthogonal projector; projects to column space of $\mathbf{B}$;

$P_B^{\perp}$ = orthogonal projector; projects to null space of $\mathbf{B}^T$;

$\mathbf{x}_{[k-1]}$ = vector containing nonnegative variables of parameters and nonnegative variables corresponding to residuals $[\mathbf{e}_1^T,\ldots,\mathbf{e}_{k-1}^T]^T$;

$(\mathbf{u}_k,\mathbf{v}_k)$ = nonnegative variables corresponding to residuals vector $\mathbf{e}_k$; and

$(\boldsymbol{\lambda},\boldsymbol{\gamma})$ = nonnegative variables corresponding to parameters vector $\boldsymbol{\beta}$.

References

Amiri-Simkooei, A. R. (2003). "Formulation of L1 norm minimization in Gauss-Markov models." *J. Surv. Eng.*, 129(1), 37–43.

Baarda, W. (1968). *A testing procedure for use in geodetic networks*, Publication on Geodesy, Netherlands Geodetic Commission, Delft, The Netherlands.

Barrodale, I., and Roberts, F. (1972). "Solution of an overdetermined equations in the L1 norm." *Mathematics Rep. No. 69*, University of Victoria, Victoria, B.C., Canada.

Barrodale, I., and Roberts, F. (1973). "An improved algorithm for discrete L1 linear approximation." *SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal.*, 10(5), 839–848.

Baselga, S. (2007). "A global optimization solution of robust estimation." *J. Surv. Eng.*, 133(3), 123–128.

Bazaraa, M. S., Jarvis, J. J., and Sherali, H. D. (1990). *Linear programming and network flows*, 2nd Ed., Wiley, New York.

Dantzig, G. B. (1963). *Linear programming and extensions*, Princeton University Press, Princeton, N.J.

Dantzig, G. B., and Thapa, M. N. (1997). *Linear programming 1: Introduction*, Springer, New York.

Hekimoglu, S. (1997). "The finite sample breakdown points of conven-

tional iterative outlier detection procedures." *J. Surv. Eng.*, 123(1), 15–31.
Junhuan, P. (2005). "The asymptotic variance-covariance matrix, Baarda test and reliability of L1 norm estimates." *J. Geodyn.*, 78, 668–682.
Koch, K. R. (1999). *Parameter estimation and hypothesis testing in linear models*, Springer, Berlin.
Krarup, T., Juhl, J., and Kubik, K. (1980). "Götterdämmerung over least squares adjustment." *Proc., 14th Congress of the Int. Society of Photogrammetry*, Vol. B3, 369–378.
Maronna, R. A., Martin, R. D., and Yohai, V. J. (2006). *Robust statistics: Theory and methods*, Wiley series in probability and statistics, U.K.
Marshall, J., and Bethel, J. (1996). "Basic concepts of L1 norm minimization for surveying applications." *J. Surv. Eng.*, 122(4), 168–179.
Pope, A. J. (1976). "The statistics of residuals and the detection of outliers." *NOAA Technical Rep. No. NOS65 NGS1*, U.S. Department of Commerce, National Geodetic Survey, Rockville, Md.
Portnoy, S., and Koenker, R. (1997). "The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators." *Stat. Sci.*, 12(4), 279–300.
Roos, C., Terlaky, T., and Vial, J. P. (2005). *Interior point methods for linear optimization*, Revised Ed., Springer, New York.
Teunissen, P. J. G., Simons, D. G., and Tiberius, C. C. J. M. (2005). "Probability and observation theory." *Lecture notes AE2-E01*, Delft University of Technology, Delft, The Netherlands.
Xu, G. (2007). *GPS: Theory, algorithms, and applications*, 2nd Ed., Springer, Berlin.
Yang, Y. (1994). "Robust estimation for dependent observations." *Manuscr. Geod.*, 19, 10–17.
Yang, Y. (1999). "Robust estimation of geodetic datum transformation." *J. Geodyn.*, 73, 268–274.