Acta Math. Univ. Comenianae
Vol. LXXXIV, 2 (2015), pp. 297–308

TWO APPROACHES FOR SOLVING l_1-REGULARIZED LEAST SQUARES WITH APPLICATION TO TRUSS TOPOLOGY DESIGN

R. KUKUMBERG

Abstract. Many real-world problems may lead to minimization of a non-differentiable convex function of a large number of variables. In this paper, we study two different approaches for solving the so-called l_1-regularized least squares problem. We apply and compare two competing methods of convex optimization to solving this problem, namely the proximal gradient method and the interior-point method. We describe two specialized inexact interior point methods for solving the l_1-regularized least squares problem and compare them with three different versions of the proximal gradient method known from literature. We illustrate performance of these methods on a truss topology design problem with more than 35 000 variables. Both methods are compared, analyzed and a discussion on the performance is provided.

1. Introduction

Consider a linear model of the form

$$b = Ax + v,$$

where $A \in \mathbb{R}^{m\times n}$ is a data matrix, $b \in \mathbb{R}^m$ is a vector of observations, $x \in \mathbb{R}^n$ is a vector of unknowns and $v \in \mathbb{R}^m$ is a noise vector. When $m < n$, the matrix A is under-determined and a simple least-squares regression leads to over-fit. A standard statistical technique against over-fitting is to include a regularization term in the least squares problem objective. Instead of the standard Tikhonov (l_2) regularization we focus on the so-called l_1-regularization to obtain an optimization problem of the following form

$$\min_x \|Ax - b\|_2^2 + \lambda\|x\|_1. \tag{1}$$

The (1) is called the l_1-regularized least squares (l_1RLS), where $\lambda > 0$ is a positive regularization parameter. It is an unconstrained convex optimization problem with a non-differentiable cost function due to the presence of the l_1 term. The optimal solution of the problem (1) always exists, but there is no analytic formula for it

Received January 19, 2015; revised July 6, 2015.
2010 *Mathematics Subject Classification.* Primary 90C25; Secondary 90C90, 90C51.
Key words and phrases. l_1-regularized least squares; proximal gradient method; proximal operator; primal-dual interior point method; truss topology design.
This research was supported by the Slovak VEGA 1/2429/12 grant.

and the solution must be computed numerically. Due to presence of the l_1 norm in the objective, the l_1RLS problem produces a sparse solution vector x and the l_1RLS is also referred to as sparse least squares. The regularization parameter λ in (1) controls trade-off of sparsity versus the first part of objective.

The (1) can be found in many fields of research. In statistics and machine learning it is used for feature selection in the Lasso regression [**1**]. It also has applications in signal processing, medical imaging [**2**] and sparse design [**3**].

The main purpose of this paper is to compare two completely different approaches to solving the l_1-regularized least squares problem. The first approach is based on the first order method called the proximal gradient method. The second one is based on the primal-dual interior point method, which is the second order method. Interestingly enough, we show that the accelerated proximal gradient method performs better in comparison to the inexact interior point methods on our test problem. The next section introduces these methods and also describes how to apply these methods for solving the problem (1). In the section 3, we compare the performance of both methods on a truss topology design problem with more than 35000 variables.

2. Solution methods

Our aim is to compare two approaches to obtaining a solution to the problem (1). They differ in a way how to deal with non-differentiability of the l_1RLS cost function. The proximal gradient method is used for minimization of the original non-differentiable unconstrained optimization problem (1). In order to use the interior point method, we need to reformulate (1) as a convex quadratic problem with linear constraints.

The interior point method for solving (1) is used, e.g., in [**2**], [**5**], [**19**]. Several other approaches have been proposed for solving (1) – e.g., homotopy methods [**6**], coordinate descent methods [**3**], the SpaRSA algorithm [**7**] and primal-dual Newton Conjugate Gradients method [**20**]. The performance of first- and second-order optimization methods for l_1RLS is studied in [**18**].

2.1. Proximal gradient method

A standard approach to solving the non-differentiable problem (1) is a subgradient method [**4**]. However, these methods have several drawbacks – they are often slow and they are not descent methods. In the view of these disadvantages, more efficient methods were created to solve problems with non-differentiable convex functions of a various structure. Unconstrained optimization problems with a cost function split in the convex differentiable part and the convex non-differentiable part, lead to the so-called Proximal Gradient (PG) method. Further information on the PG method can be found in [**8**], [**9**], [**10**], [**17**].

The PG method is used for optimization of an unconstrained problem with a cost function $f(x)$ split in two components

$$\min f(x) = g(x) + h(x), \tag{2}$$

where the function $g(x)\colon \mathbb{R}^n \to \mathbb{R}$ is differentiable and convex, and the function $h(x)$ is closed, convex and possibly non-differentiable with an explicit or inexpensive proximal operator.

An essential element of the PG method is the so-called proximal operator. The proximal operator associated with the convex function $h(x)$ is defined as follows

$$\text{prox}_h(x) = \underset{u}{\text{argmin}} \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right). \tag{3}$$

It can be easily shown that the proximal operator of a convex function exists and is unique for all x. However, the proximal operator of some functions can not be expressed explicitly, what can hamper the effectiveness of the PG method.

For optimizing (2), the basic proximal gradient method has the main iteration of the form

$$x^{(k)} = \text{prox}_{t_k h}\left(x^{(k-1)} - t_k \nabla g(x^{(k-1)}) \right),$$

where $t_k > 0$ is the step size, which can be set constant or determined by a line search at each iteration. Note that if we take $h(u) \equiv 0$, we get the iteration of the gradient descent method in the form $x^{(k)} = x^{(k-1)} - t_k \nabla g(x^{(k-1)})$.

The basic PG method exhibits the convergence rate $O(1/k)$ (see [**9**], [**10**]). The convergence of the PG method is guaranteed for a constant step size and also for the step size determined by a line search under two standard assumptions:

- The optimal value of objective function f^* is finite and attained at x^* (not necessarily unique).
- The gradient ∇g is Lipschitz continous with a Lipschitz constant $L > 0$, that is

$$\|\nabla g(x) - \nabla g(y)\|_2 \leq L\|x - y\|_2, \qquad \text{for all } x, y \in \mathbb{R}^n.$$

The basic PG method is a descent method – the value of the objective function at the next iterate decreases. Moreover, the distance of the iterate from the optimal point is non-increasing in every step of the algorithm, i.e., $\|x^+ - x^*\|_2 \leq \|x - x^*\|_2$ [**17**].

2.1.1. Accelerated proximal gradient method. A convergence rate of the basic PG algorithm can be improved to $O(1/k^2)$ by extrapolation [**8**], [**10**]. The modified algorithm is called the accelerated proximal gradient (APG) method or the PG method with extrapolation. The APG has approximately the same complexity per iteration as the basic PG. The APG has main iteration of the form

$$\begin{aligned} x^{(k)} &= \text{prox}_{t_k h}\left(y^{(k-1)} - t_k \nabla g(y^{(k-1)}) \right) \\ y^{(k)} &= x^{(k)} + \frac{k-1}{k+2}(x^{(k)} - x^{(k-1)}), \end{aligned}$$

where the step size t_k is either constant or determined by a line search. Under the same assumptions as the basic method, the APG exhibits a faster convergence rate $O(1/k^2)$, which is optimal for this class of problems [**8**], [**10**]. However, the APG is not a descent method, it does not guarantee that the value of the objective is

non-increasing, it often reaches better value by 'jumps'. This fact is a motivation for other modifications of the method such as the descent version of APG.

2.1.2. Descent version of the accelerated proximal gradient method. The following modification of APG algorithm guarantees non-increasing value of the objective at the next iterate (i.e., $f(x^{(k)}) \leq f(x^{(k-1)})$):

$$\begin{aligned} z^{(k)} &= \operatorname{prox}_{t_k h}\left(y^{(k-1)} - t_k \nabla g(y^{(k-1)})\right) \\ x^{(k)} &= \begin{cases} z^{(k)} & f(z^{(k)}) \leq f(x^{(k-1)}) \\ x^{(k-1)} & \text{otherwise} \end{cases} \\ v^{(k)} &= x^{(k-1)} + \frac{1}{\theta_k}(z^{(k)} - x^{(k-1)}) \\ y^{(k)} &= (1-\theta_{k+1})x^{(k)} + \theta_{k+1} v^{(k)}, \end{aligned}$$

where parameter θ_k is defined as $\theta_k = \frac{2}{k+1}$. Note that if condition $f(z^{(k)}) \leq f(x^{(k-1)})$ does not hold, the value of variable $x^{(k)}$ is kept constant until the better suboptimal solution is found.

2.1.3. Solution of l_1RLS – PG. As the l_1RLS is an unconstrained minimization problem with the cost function split in the differentiable convex part $\|Ax-b\|_2^2$ and the non-differentiable convex part $\lambda\|x\|_1$, the proximal gradient method can be used for solving (1). The essential element of all PG algorithms, the proximal operator associated with the function $th(u) = t\lambda\|u\|_1$, is simply expressed as follows

$$\operatorname{prox}_{th}(x)_i = \begin{cases} x_i - t\lambda, & x_i > t\lambda, \\ x_i + t\lambda, & x_i < -t\lambda, \\ 0, & |x_i| \leq t\lambda, \end{cases}$$

due to separability of the function $h(u)$. An equivalent vector formulation of the proximal operator, more suitable for MATLAB implementation, is of the form

$$\operatorname{prox}_{th}(x) = \operatorname{sign}(x)\max[0, |x| - t\lambda].$$

Another important element of the PG algorithm is a step size strategy. The convergence of PG methods is guaranteed for a constant step size $t = 1/L$, where L is a Lipschitz constant of the gradient of the function $g(x) = \|Ax-b\|_2^2$. Therefore, we can set the constant step size $t = \frac{1}{2\sigma_{\max}(A^{\mathrm{T}}A)}$ in the PG algorithms, where $\sigma_{max}(A^{\mathrm{T}}A)$ is the largest eigenvalue of the matrix $A^{\mathrm{T}}A$.

2.2. Primal-dual interior point method

Recently, interior point methods are widely used for solving convex programming problems with differentiable functions. In this section, we focus on a primal-dual interior point method (PDIPM). For more details on these methods, we refer the reader to [**11**], [**12**], [**13**].

The PDIPM solves the problem by applying Newton's method to a sequence of modified Karush-Kuhn-Tucker (KKT) conditions, where the complementary

slackness condition is perturbed by a positive parameter. The PDIPM updates both primal and dual variables at each iteration.

Consider the following optimization problem

$$\begin{aligned} &\min \ f_0(x) \\ &\text{s.t. } f_i(x) \le 0, \quad i = 1, \dots, m, \end{aligned} \tag{4}$$

where $f_0, \cdots, f_m \colon \mathbb{R}^n \to \mathbb{R}$ are convex, twice continuously differentiable functions. The direction of optimization of primal and dual variables, primal-dual search direction, is obtained from Newton's method applied to the nonlinear system of equations

$$r_t(x, \mu) = \begin{pmatrix} \nabla f_0(x) + Df(x)^{\mathrm{T}} \mu \\ -\operatorname{diag}(\mu) f(x) - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0, \tag{5}$$

which represents the modified KKT equations, where the complementary slackness condition is perturbed by a positive term $\frac{1}{t}$. Here $f = (f_1, \dots, f_m)^{\mathrm{T}}$ and $Df(x)$ is the Jacobian matrix. The primal-dual search direction $\Delta y = (\Delta x, \Delta \mu)$ is then defined as a solution of the system of linear equations

$$Dr_t(y) \Delta y = -r_t(y), \tag{6}$$

where $Dr_t(y)$ is the Jacobian. As $t \to \infty$, system (5) reduces to the original KKT condition for (4). Therefore, the basic idea is to take PD search directions for a sequence of increasing values of t. This leads to the following algorithm.

PDIPM Algorithm

Input: $x : f(x) < 0$; $\mu > 0$; $\nu > 1$; $\varepsilon > 0$, $\varepsilon_f > 0$

Repeat steps 1, 2, 3, 4, 5, **until** $\|r_1\|_2 \le \varepsilon_f$ and $\hat{\eta} \le \varepsilon$.

1. Set $t = \nu m / \hat{\eta}$.
2. Compute PD search direction $\Delta y_{pd} = (\Delta x_{pd}, \Delta \mu_{pd})$.
3. Determine step size $s > 0$ by line search. Set

$$s = 0.99 \sup\{s \in [0, 1] | \mu + s \Delta \mu \ge 0\}$$

and multiply s by $\beta \in (0, 1)$ until

$$f(x^+) < 0, \qquad \|r_t(x^+, \mu^+)\|_2 \le (1 - \alpha s) \|r_t(x, \mu)\|_2.$$

4. Update the iterate by $y = y + s \Delta y_{pd}$.
5. Compute the duality gap $\hat{\eta}(x, \mu) = -f(x)^{\mathrm{T}} \mu$.

2.2.1. Solution of l_1RLS – PDIPM. Interior point methods are widely used for solving differentiable convex programming problems. To get rid of non-differentiability of the l_1RLS (1) and to transform (1) to a problem solvable by the PDIPM, we have to introduce new variables and new constraints. There are two main options.

Option 1. The l_1RLS (1) can be transformed to equivalent formulation by introducing a new variable $u \in \mathbb{R}^n$ and new constraints $-u_i \le x_i \le u_i, \quad i = 1, \dots, n$,

$$\begin{aligned} \min_{x,u} \ & \|Ax - b\|_2^2 + \lambda \mathbf{1}^{\mathrm{T}} u \\ \text{s.t.} \ & x - u \le 0, \\ & -x - u \le 0, \end{aligned} \tag{7}$$

where $\mathbf{1}$ denotes a vector of ones. Formulation (7) is a convex quadratic problem, with a differentiable quadratic objective function and linear constraints. Note that the number of variables is doubled by this transformation.

In the case of the option 1, the primal-dual search direction is the Newton step for the system of nonlinear equations

$$r_t(x, u, \mu_1, \mu_2) \equiv \begin{pmatrix} 2A^{\mathrm{T}}(Ax - b) + \mu_1 - \mu_2 \\ \lambda \mathbf{1} - \mu_1 - \mu_2 \\ -\operatorname{diag}(\mu_1)(x - u) - \frac{1}{t}\mathbf{1} \\ -\operatorname{diag}(\mu_2)(-x - u) - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0.$$

The PD search direction $(\Delta x, \Delta u, \Delta \mu_1, \Delta \mu_2)$ is then defined as a solution of the system of linear equations (6), which can be written as

$$\left(\begin{array}{cccc|c} 2A^{\mathrm{T}}A & 0 & I & -I & -p - \mu_1 + \mu_2 \\ 0 & 0 & -I & -I & -\lambda \mathbf{1} + \mu_1 + \mu_2 \\ -M_1 & M_1 & U - X & 0 & M_1(x - u) + \frac{1}{t}\mathbf{1} \\ M_2 & M_2 & 0 & X + U & M_2(-x - u) + \frac{1}{t}\mathbf{1} \end{array} \right),$$

where $X = \operatorname{diag}(x)$, $U = \operatorname{diag}(u)$, $M_1 = \operatorname{diag}(\mu_1)$, $M_2 = \operatorname{diag}(\mu_2)$ and $p = 2A^{\mathrm{T}}(Ax - b)$. Since solving this whole system is not effective, we have to reduce it. By denoting $J_1 = M_1^{-1}(X - U)$, $J_2 = M_2^{-1}(-X - U)$ and eliminating $(\Delta u, \Delta \mu_1, \Delta \mu_2)$, we obtain a reduced system of the form

$$(2A^{\mathrm{T}}A - J_p + J_m J_p^{-1} J_m)\Delta x = -p + s_1 - s_2 - J_m J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2), \tag{8}$$

where $J_p = J_1^{-1} + J_2^{-1}$, $J_m = J_1^{-1} - J_2^{-1}$ are diagonal matrices and $s_1 = \frac{1}{t}(X - U)^{-1}\mathbf{1}$, $s_2 = \frac{1}{t}(-X - U)^{-1}\mathbf{1}$ are vectors. Clearly, the matrix $(2A^{\mathrm{T}}A - J_p + J_m J_p^{-1} J_m) \in \mathbb{R}^{n \times n}$ from system (8) is symmetric and positive definite.

The other three components of the PD search direction Δu, $\Delta \mu_1$, $\Delta \mu_2$ can be computed as follows

$$\begin{aligned} \Delta u &= J_p^{-1}(\lambda \mathbf{1} + s_1 + s_2 + J_m \Delta x) \\ \Delta \mu_1 &= -\mu_1 - s_1 + J_1^{-1}(-\Delta x + \Delta u) \\ \Delta \mu_2 &= -\mu_2 - s_2 + J_2^{-1}(\Delta x + \Delta u), \end{aligned} \tag{9}$$

where matrices J_1, J_2, J_p, J_m are diagonal ones. Therefore, the computation cost of components Δu, $\Delta \mu_1$, $\Delta \mu_2$ is low.

Option 2. The l_1RLS (1) can be transformed to a convex quadratic problem also by splitting the variable x into positive and negative parts. We replace x by $x = x^+ - x^-$, where $x^+ = \max(0, x)$ and $x^- = \max(-x, 0)$. Then $\|x\|_1 =$

$\|x^+ - x^-\|_1 = \mathbf{1}^{\mathrm{T}}x^+ + \mathbf{1}^{\mathrm{T}}x^-$. By this transformation, we obtain an equivalent formulation of the form

$$\begin{aligned} &\min \|\tilde{A}z - b\|_2^2 + \lambda \mathbf{1}^{\mathrm{T}} z, \\ &\text{s.t. } z \geq 0 \end{aligned} \tag{10}$$

where $\tilde{A} = [A, -A] \in \mathbb{R}^{m\times 2n}$ and $z = \begin{pmatrix} x^+ \\ x^- \end{pmatrix} \in \mathbb{R}^{2n}$ is a new variable. As in the first option, we obtain a problem with a quadratic objective function and linear constraints and the number of variables is also doubled.

For (10), the primal-dual search direction is obtained from Newton's method applied to the system of nonlinear equations

$$r_t(z,\mu) = \begin{pmatrix} 2\tilde{A}^{\mathrm{T}}(\tilde{A}z - b) + \lambda\mathbf{1} - \mu \\ \operatorname{diag}(\mu)z - \frac{1}{t}\mathbf{1} \end{pmatrix} = 0.$$

The PD search direction $(\Delta z, \Delta\mu)$ is then defined as a solution to the system of linear equations (6), which is of the form

$$\left(\begin{array}{cc|c} 2\tilde{A}^{\mathrm{T}}\tilde{A} & -I & -2\tilde{A}^{\mathrm{T}}(\tilde{A}z - b) - \lambda\mathbf{1} + \mu \\ \operatorname{diag}(\mu) & \operatorname{diag}(z) & -\operatorname{diag}(\mu)z + \frac{1}{t}\mathbf{1} \end{array}\right).$$

By eliminating $\Delta\mu$, we obtain the reduced system

$$(2\tilde{A}^{\mathrm{T}}\tilde{A} + J)\Delta z = -2\tilde{A}^{\mathrm{T}}(\tilde{A}z - b) - \lambda\mathbf{1} + s, \tag{11}$$

where $s = \frac{1}{t}\operatorname{diag}(z)^{-1}\mathbf{1}$ and $J = \operatorname{diag}(\mu)\operatorname{diag}(z)^{-1}$ is a diagonal matrix. The matrix $(2\tilde{A}^{\mathrm{T}}\tilde{A} + J) \in \mathbb{R}^{2n\times 2n}$ in this system is symmetric and positive definite. The second component of PD search direction $\Delta\mu$ can be easily obtained as follows

$$\Delta\mu = -J\Delta z - \mu + s. \tag{12}$$

2.2.2. Solving Newton systems. Each iteration of the PDIPM is dominated by the cost of computing the PD search direction from system (8) or (11). For the l_1RLS (1) of large dimensions, solving these systems exactly is not computationally efficient. However, we do not need to solve these systems exactly. Instead of direct methods, we can use preconditioned iterative methods to obtain an approximate solution to system (8) or (11). For more details on inexact interior point methods, the reader is referred to, e.g., [**2**], [**5**], [**14**], [**15**], [**19**].

In our experiments, we compute the PD search direction approximately, applying the preconditioned minimum residual method (Minres). The Minres method requires the system matrix to be symmetric but need not to be positive definite. This condition is satisfied for both systems (8) and (11). The Minres algorithm uses a preconditioner P, which has to be symmetric and positive definite. We set $P = kI - J_p + J_m J_p^{-1} J_m$ for system (8) and $P = kI + J$ for (11), where k is a positive parameter. As an initial point for the Minres algorithm, we use the search direction found in the previous step of the PDIPM.

3. Numerical experiments – Truss Topology Design

One of many applications of the l_1-regularized least squares is truss topology design (TTD) [**3**], [**16**]. Truss is a mechanical construction made of elastic bars linked to each other at nodes, which can be fixed or free. We can consider structures like electric masts, railroad bridges or the Eiffel tower as trusses. The truss can be exposed to an external load, under which construction is deformed. The goal of TTD is for a given nodal grid and forces acting at the nodes to construct a truss of maximum rigidity.

In the simple TTD problem, we are given an $r \times c$ grid of nodes, $m/2$ of the nodes are free, others are fixed. Pairs of nodes can be linked by a bar. By n, we denote the number of potential bars. Note that as the grid gets larger, dimensions of the TTD problem ($A \in \mathbb{R}^{m\times n}$) considerably increase. By a vector $b \in \mathbb{R}^m$, we denote 2D external forces acting at the free nodes. Our aim is to construct such a structure that withstands these forces in an optimal way. An attractive feature of the TTD problem is such that although it seems to deal with the weights of the bars only, it finds the geometric shape of the truss as well. In the optimal truss, majority of the bars will have zero weights, which is guaranteed by the property of the l_1 regularization.

3.1. Experiment 1 – Comparison of methods

In this section, we compare the performance of our MATLAB implementations of 3 versions of the PG method (basic, Accelerated and Descent PG) and 2 versions of the PDIPM (one for reformulation (7) and other one for (10)). All experiments were performed in MATLAB running on a notebook with Intel Core-i7 2.4GHz processor and 8GB RAM.

We illustrate the performance of methods on a TTD problem with more than 35300 variables/potential bars. Our test problem is a bridge type TTD problem with grid size 7×49 and matrix $A \in \mathbb{R}^{678\times 35382}$. In TTD, the matrix A is sparse, it has at most 4 nonzeros per column (in 2D truss). There are 4 fixed nodes spaced at the bottom (representing the pillars of the bridge) and a unit downward force is applied at every node at the height 1. An initial point is set to be the vector of zeros and the regularization parameter is set $\lambda = 0.0002$ for every method. In tests, we give each method the same time for solving the TTD problem (160s, 300s, 700s) and we compare the number of iterations, duality gaps and trusses attained by each method.

In Table 1, we can see the suboptimality and the number of iterations attained by each method after 160, 300 and 700 seconds. As the measure of suboptimality, we use the dual gap between the primal objective attained by the method and the fixed chosen value of dual objective. It is easily observable that the proximal gradient methods, which are the first order methods, perform far more iterations than primal-dual interior point methods for the same given time. For example, the APG method does over 1.4 million iterations after 700 seconds, whereas the PDIPM does only 69. Note that the APGM does twice as many iterations as the DPGM, mainly because the DPGM evaluates the objective function in every

time	160s		300s		700s	
	# it.	value	# it.	value	# it.	value
PGM	361216	6.25e-04	677245	4.61e-04	1554266	3.04e-04
APGM	326247	**9.68e-08**	609655	**1.98e-10**	1429618	**1.00e-10**
DPGM	156707	2.87e-07	290189	1.29e-07	724994	**1.00e-10**
PDIPM1	26	7.92e-05	36	6.07e-07	69	1.27e-09
PDIPM2	19	7.04e-04	24	1.34e-04	39	9.07e-07

Table 1. Comparison of methods – the number of iterations and suboptimality.

step and this can be prohibitively expensive for large scale problems. We also observe that the PDIPM for an equivalent formulation (7) performs better than the PDIPM for (10). Main advantage of the PDIPM1 is that the dimension of the linear system in computation of the PD search direction (8) is $n \times n$ whereas the dimension of (11) is $2n \times 2n$. Even though the PG method performs most iterations, its rate of convergence is not good enough to compete with the accelerated and the descent version of the algorithm or the PDIPMs. Overall, the APGM converges fastest to an optimal solution and attains best suboptimality at each time. A slightly worse performance is provided by the DPGM and the PDIPM1. Despite of slower theoretical convergence, the APGM (first order method) performs better than interior point methods thanks to the low computational cost of its iterations. Note that this is the case because of the size of the problem (and the sensitivity of IPMs to it) – for a smaller grid size IPMs should perform better than PG methods.

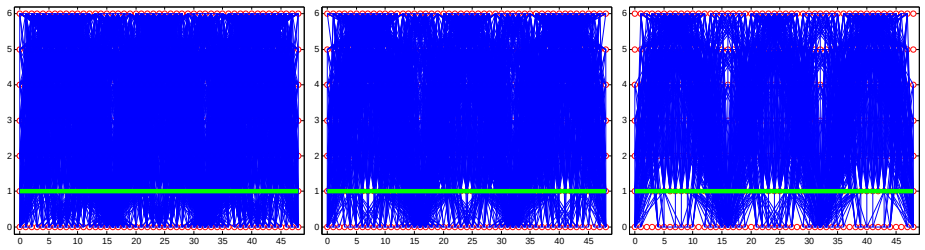

Figure 1. Computational results for proximal gradient method (PGM) for termination times 160, 300 and 700 seconds.

When we compare plotted suboptimal solutions (trusses in Figures 1, 2, 3, 4, 5) by methods after 160s, 300s, 700s, we can see similar behavior as in Table 1. Pictures of solution by the PGM do not resemble bridges. However, the APGM and the DPGM do produce visibly bridge-like structures even after 160 seconds and we can observe pillar positions. Majority of the bars gets zero weight. This observation is not surprising, because the attained dual gaps are better than 3e-07. From the last pictures, we can see that the PDIPM1 perform much better than

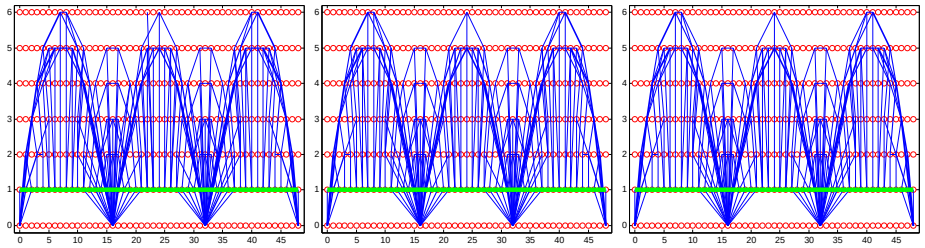

Figure 2. Computational results for accelerated proximal gradient method (APGM) for termination times 160, 300 and 700 seconds.

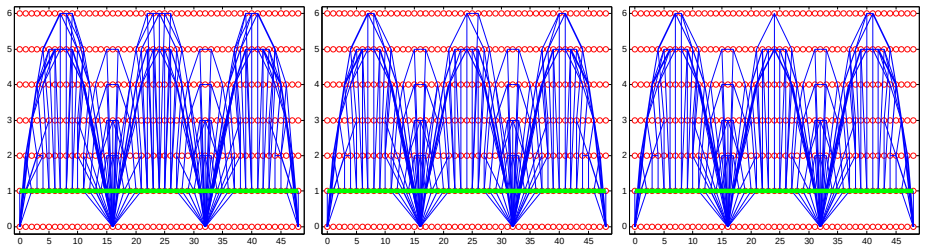

Figure 3. Computational results for descent version of accelerated proximal gradient method (DPGM) for termination times 160, 300 and 700 seconds.

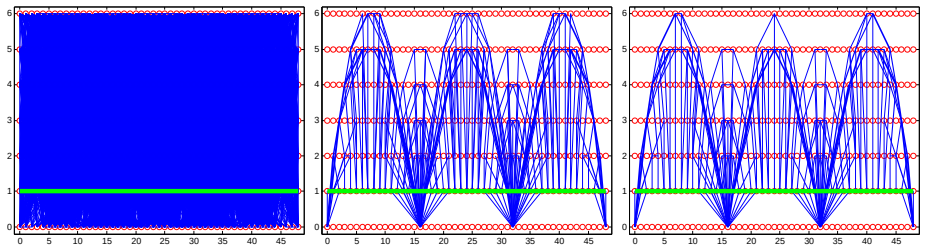

Figure 4. Computational results for primal-dual interior point method for formulation (7) (PDIPM1) for termination times 160, 300 and 700 seconds.

the PDIPM2. Again, we can consider the APGM as winner of this comparison. Final trusses of the APGM, the DPGM and the PDIPM1 are quite comparable.

3.2. Experiment 2 – Number of pillars

In the second experiment, we study impact of the number of fixed nodes (pillars) on the geometric shape of a truss. Our test problem is a bridge type TTD problem with the grid size 7×40 and over 23500 potential bars (variables). Again, a unit

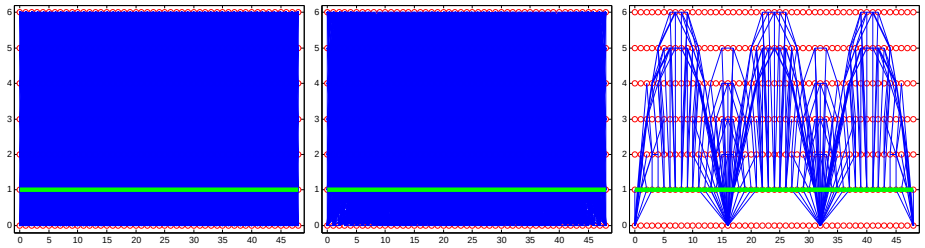

Figure 5. Computational results for primal-dual interior point method for formulation (10) (PDIPM2) for termination times 160, 300 and 700 seconds.

downward force is applied at every node at the height 1. An initial point is set to be the zero vector and $\lambda = 0.0002$. We compare trusses with 3, 4 and 6 pillars. The TTD problems are solved by the PDIPM1 with the same desired accuracy.

From Figure 6, we can see that when the number of pillars increases, then height of the construction decreases. We also observe that more bars are needed when truss has 3 pillars in comparison with 6 pillars. The value of the objective function decreases with the number of pillars as 4.05e-03, 2.64e-03, 1.65e-03. Therefore, from economic point of view, it is better to use more pillars to save money for material. Note that if we allow to move these pillars (optimal placement of pillars), it is possible to reach even a better value of the objective function.

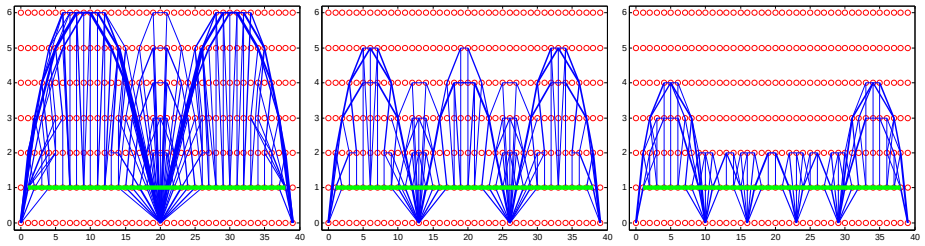

Figure 6. Number of pillars – 3, 4, 6.

4. Conclusion

Our paper dealt with the famous l_1-regularized least squares problem. We analyzed and compared two different approaches for solving this problem. The first approach was based on application of proximal gradient methods, which are the first order methods, applied to an original formulation of l_1RLS problem. In the second approach, we described two specialized inexact primal-dual interior point methods, which used a preconditioned iterative method for finding a primal-dual search direction. Interior point methods were applied to equivalent differentiable formulations of the original problem. We illustrated and discussed the performance of these approaches on a truss topology design problem with over 35000 variables.

Despite of slower theoretical convergence, the accelerated PG performed better than the interior point methods thanks to low computational cost of its iterations. All methods except of basic PG produced visibly bridge-like structure/solution after 700 seconds.

References

1. Tibshirani R., *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc., Ser. B **58(1)** (1996), 267–288.
2. Kim S.-J., Koh K., Lustig M., Boyd S. and Gorinevsky D., *An interior-point method for large-scale l1-regularized least squares*, Selected Topics in Signal Processing, IEEE Journal of, **1(4)** (2007), 606–617.
3. Richtarik P. and Takac M., *Efficient serial and parallel coordinate descent methods for huge-scale truss topology design*, Operations Research Proceedings 2011, Springer Berlin Heidelberg, 2012, 27–32.
4. Polyak B., *Introduction to optimization. (Vvedenie v optimizatsiyu).*, Nauka, Moskva 1983.
5. Candes E. and Romberg J., *l1-MAGIC: Recovery of Sparse Signals via Convex Programming*, A Collection of MATLAB Routines for Solving the Convex Optimization Programs Central to Compressive Sensing, 2005.
6. Donoho D. and Tsaig Y., *Fast solution of l_1-norm minimization problems when the solution may be sparse*, IEEE Trans. Inf. Theory **54(11)** (2008), 4789–4812.
7. Wright S. J., Nowak R. D. and Figueiredo M. A., *Sparse reconstruction by separable approximation*, Signal Processing, IEEE Transactions on, **57(7)** (2009), 2479–2493.
8. Beck A. and Teboulle M., *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci. 2(1) (2009), 183–202.
9. Parikh N. and Boyd S., *Proximal algorithms*, Foundations and Trends in Optimization, **1(3)** (2013), 123–231.
10. Nesterov Y., *Gradient methods for minimizing composite objective function*, Mathematical Programming, **140(1)**, (2007), 125–161.
11. Boyd S. and Vandenberghe L., *Convex optimization*, Cambridge University Press, 2004.
12. Wright S. J., *Primal-dual interior-point methods*, Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics xx, 1997.
13. Nesterov Y., Nemirovskii A. and Ye Y., *Interior-point polynomial algorithms in convex programming*, SIAM Studies in Applied Mathematics. 13. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. ix, 1994.
14. Koh K., Kim S. J. and Boyd S., *An interior-point method for large-scale l1-regularized logistic regression*, J. Mach. Learn. Res. **8** (2007), 1519–1555.
15. Gondzio J., *Matrix-free interior point method*, Comput. Optim. Appl. **51(2)** (2012), 457–480.
16. Ben-Tal A. and Nemirovski A., *Lectures on modern convex optimization. Analysis, algorithms, and engineering applications.*, PA: SIAM, PA: MPS, 2001.
17. Vandenberghe L., *EE236C – Optimization Methods for Large-Scale Systems*, lectures, 2013, `http://www.seas.ucla.edu/~vandenbe/ee236c.html`
18. Fountoulakis K. and Gondzio J., *Performance of First-and Second-Order Methods for Big Data Optimization*, arXiv preprint arXiv:1503.03520, (2015).
19. Fountoulakis K., Gondzio J. and Zhlobich P., *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, **6(1)** (2014) 1-31.
20. Fountoulakis K. and Gondzio J., *A second-order method for strongly convex l1-regularization problems*, Mathematical Programming, 1–31, 2013.

R. Kukumberg, Department of Applied Mathematics and Statistics, FMFI UK, Mlynska Dolina, 842 48 Bratislava, Slovakia, *e-mail*: `kukumberg.roman@gmail.com`