Primal-dual subgradient method for Huge-Scale Linear Conic Problems

Yu. Nesterov * S.Shpirko†

August, 2012

Abstract

In this paper we develop a *primal-dual* subgradient method for solving huge-scale Linear Conic Optimization Problems. Our main assumption is that the primal cone is formed as a direct product of many small-dimensional convex cones, and that the matrix A of corresponding linear operator is *uniformly sparse*. In this case, our method can approximate the primal-dual optimal solution with accuracy ϵ in $O\left(\frac{1}{\epsilon^2}\right)$ iterations. At the same time, complexity of each iteration of this scheme does not exceed $O(rq \log_2 n)$ operations, where r and q are the maximal numbers of nonzero elements in the rows and columns of matrix A, and n is the number variables.

Keywords: Convex optimization, subgradient methods, huge-scale problems, sublinear iteration cost.

*Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium; e-mail: nesterov@core.ucl.ac.be.

The research results presented in this paper have been supported by a grant "Action de recherche concertè ARC 04/09-315" from the "Direction de la recherche scientifique - Communautè française de Belgique". The scientific responsibility rests with its author(s).

†Moscow Institute of Physics and Technology. E-mail: shpirko@yahoo.com. The research presented in this paper was supported by the Laboratory of Structural Methods of Data Analysis in Predictive Modeling, MIPT, through the RF government grant, ag.11.G34.31.0073

1 Introduction

1. Motivation. Development of the new computer technologies in the last decade resulted in an increasing interest to optimization problems of extremely big size, with millions and billions of variables. The main sources of such problems are Internet and Telecommunications. However, even in more traditional applications (Finite Elements Approximations, Partial Differential Equations), we always can create optimization problems of extremely big dimension. For problems of this type (we call them *Huge-Scale Optimization Problems*), even the simplest vector operations become very expensive. They can be solved only by methods with very sophisticated treatment of corresponding sparsity patterns.

In the last years we can see a revival of interest to the coordinate-descent methods [5, 2, 4]. These methods benefit from a very simple iteration, which usually has a sublinear computational cost. However, from the view point of iteration complexity, their convergence rate is lower than that of the standard gradient methods.

Recently, in [1] it was shown that for a special class of nonsmooth optimization problems it is possible to reach sublinear iteration complexity without reducing the optimal rate of convergence of the simplest subgradient schemes. The functions from this problem class are characterized by *sparse subgradients*. At each iteration, this feature allows to update only a few entries of the test points. Consequently, it is cheap to *update* the results of corresponding matrix/vector products (for sparse matrices). Another important element in this technique is a short Binary Tree Table, which allows efficient recomputation of the maximum of n elements after changing a single entry in the array (it needs $\log_2 n$ operations).

For problems with an appropriate sparsity pattern, the methods presented in [1] have *logarithmic* dependence of the complexity of the usual subgradient iteration on the dimension of the space of variables. This allows to use the proposed technique for solving optimization problems of practically unlimited size.

In paper [1] there were presented three different methods for solving a structural *primal* problem. However, in many situation it is interesting to approximate the optimal dual variables. This is the main goal of our paper.

In this paper we develop a *primal-dual* subgradient method for solving a huge-scale *Linear Conic Optimization Problem*. Traditionally, the problems of this type are treated by the Interior-Point Methods [3]. However, because of the memory limitations and cubic complexity of each iteration, these methods can be used only for solving problems of moderate size. Our method can solve extremely big primal-dual linear conic problems

$$\min_{x \in K} \{\langle c, x \rangle : \; Ax = b\} \; = \; \max_{\substack{s \in K^*, \\ y \in R^m}} \{\langle b, y \rangle : \; s + A^* y = c\}, \tag{1.1}$$

where K is a closed convex cone, K^* is the cone dual to K, and the operator A^* is adjoint to A. Our main assumption is that K is formed as a direct product of many small-dimensional convex cones, and that the matrix of the linear operator A is uniformly sparse (see Section 3 for details). In this case, our method can approximate the primal-dual solution of problem (1.1) with accuracy ϵ in $O\left(\frac{1}{\epsilon^2}\right)$ iterations. At the same time, the complexity of each iteration of this scheme depends *logarithmically* on the dimension of the space of variables.

2. Contents. In Section 2, we describe the structure of our problem and introduce the main assumptions. Our method generates the minimization sequence for the dual problem in (1.1), approximating at the same time the optimal primal solution. In order to apply our technique, we need to rewrite the conic constraints of the dual problem in a functional form.

In our scheme, the generated primal solutions are always feasible for the cone K, and the dual solutions always satisfy dual equality constraints. They are approximately feasible for the dual cone K^*. The level of dual infeasibility depends on the step-size parameter h, which bounds also the primal-dual gap. We prove that in the limit the residual of the primal system vanishes. The worst-case complexity bound for our method is $O\left(\frac{1}{\epsilon^2}\right)$ iterations, where ϵ is the desired accuracy of the primal-dual solution. In Section 3 we give an example of sparsity pattern for the problem data, which results in logarithmic dependence of the cost of each iteration of our scheme on the dimension of the space of variables. In particular, this happens when the matrix A has only few nonzero diagonals. (This is typical for Finite-Element Approximations.) In the last Section 4 we present preliminary computational experiments with our scheme as applied to Truss Topology Design problems.

3. Notation and generalities. We denote by E a finite-dimensional linear space, and by E^* the dual space formed by all linear functions on E. The value of function $s \in E^*$ at $x \in E$ is denoted by $\langle s, x \rangle$. For a linear operator $A : E_1 \to E_2^*$ we define the *adjoint operator* $A^* : E_2 \to E_1^*$ in the standard way:

$$\langle Ax, y \rangle \;=\; \langle A^* y, x \rangle, \quad x \in E_1,\ y \in E_2.$$

In the special case $E = E^* = R^m$, we have

$$\langle x, y \rangle \;=\; \sum_{i=1}^{m} x^{(i)} y^{(i)}, \quad x, y \in R^m.$$

For $y \in R^m$ we always use the Euclidean norm $\|y\| = \langle y, y \rangle^{1/2}$. Operator $A : R^n \to R^m$ is identified with $m \times n$-matrix, In this case, $A^* \equiv A^T$.

A closed convex pointed cone $K \subset E$ with nonempty interior is called *normal*. For such a cone, the *dual cone* $K^* = \{s \in E^* : \langle s, x \rangle \geq 0\}$ is also normal.

For normal cone K, we can always point out a logarithmically homogeneous *self-concordant barrier* $F(x)$:

$$F(tx) \;=\; F(x) - \nu \ln t, \quad x \in \operatorname{int} K,\ t > 0,$$

where ν is called the *parameter* of the barrier [3]. Then the dual barrier

$$F_*(s) \;=\; \max_{x \in \operatorname{int} K} [-\langle s, x \rangle - F(x)], \quad s \in \operatorname{int} K^*,$$

is logarithmically homogeneous self-concordant barrier for K^*. For $x \in \operatorname{int} K$ we have $-\nabla F(x) \in \operatorname{int} K^*$. Moreover, we have the following useful identities:

$$-\nabla F_*(-\nabla F(x)) \;=\; x, \quad \nabla^2 F_*(-\nabla F(x)) \;=\; [\nabla^2 F(x)]^{-1}, \quad x \in \operatorname{int} K. \tag{1.2}$$

For $h \in K$ and $x \in \operatorname{int} K$, by Theorem on Recession Direction [3] we have

$$\langle \nabla^2 F(x) h, h \rangle \;\leq\; \langle \nabla F(x), h \rangle^2. \tag{1.3}$$

2 Primal-dual subgradient method for Linear Conic Problems

In this section we consider Linear Conic Optimization problems. Assume that the space of primal variables E is partitioned as follows:

$$x^j \in E_j,\ j = 1, \ldots, n, \quad x = (x^1, \ldots, x^n) \in E, \tag{2.1}$$

where E_j are some finite-dimensional spaces. Thus, $\dim E = \sum\limits_{j=1}^{n} \dim E_j$, and $\langle c, x \rangle \stackrel{\text{def}}{=} \sum\limits_{j=1}^{n} \langle c^j, x^j \rangle$ for any $c \in E^*$. For a linear operator $A : E \to R^m$ we use corresponding partition:

$$A = (A_1, \ldots, A_n), \quad Ax \stackrel{\text{def}}{=} \sum_{j=1}^{n} A_j x^j,\ x \in E. \tag{2.2}$$

For a conic constraint $x \in K$, we assume that $K = \bigotimes\limits_{j=1}^{n} K_j$, where all cones $K_j \subset E_j$ are closed convex and pointed. Thus, $K^* = \bigotimes\limits_{j=1}^{n} K_j^*$.

Consider the following primal Linear Conic Problem:

$$f_* \stackrel{\text{def}}{=} \inf_{x \in K} \{ \langle c, x \rangle :\ Ax = b \}. \tag{2.3}$$

Its formally dual problem can be written as follows:

$$\sup_{y \in R^m,\ s \in K^*} \{ \langle b, y \rangle :\ s + A^* y = c \}. \tag{2.4}$$

Our main assumption is as follows:

$$\textit{Dual Problem (2.4) is solvable.} \tag{2.5}$$

Denote by y_* one of its optimal solution, and by $s_* \stackrel{\text{def}}{=} c - A^* y_* \in K^*$ the corresponding slack variable. By Duality Theorem (see T.4.2.1 in [3]) and (2.5), the primal problem (2.3) is also solvable and for the primal-dual pair (2.3), (2.4) there is no duality gap:

$$\langle s^*, x^* \rangle = 0. \tag{2.6}$$

Let us treat the constraints of the dual problem in a separable form:

$$\sup_{y \in R^m,\ s \in E^*} \left\{ \langle b, y \rangle :\ s^j = c^j - A_j^* y \in K_j^*,\ j = 1, \ldots, n \right\}. \tag{2.7}$$

We need to rewrite these constraints in a functional form. For this purpose, in each cone K_j^* we fix a *scaling element* $d^j \in \operatorname{int} K_j^*$, $j = 1, \ldots, n$. Then, for $u^j \in E_j^*$, we can define the following function:

$$\psi_j(u^j) \stackrel{\text{def}}{=} \min_{\tau} \{ \tau :\ \tau d^j - u^j \in K_j^* \}. \tag{2.8}$$

Lemma 1 *Function $\psi_j(u^j)$ is convex on E_j. It has the following representation:*

$$\psi_j(u^j) \;=\; \max_{x^j \in K_j} \{\langle u^j, x^j\rangle : \; \langle d^j, x^j\rangle = 1\}. \tag{2.9}$$

Thus, $\partial\psi_j(u^j) = \mathrm{Arg}\max_{x^j \in K_j} \{\langle u^j, x^j\rangle : \; \langle d^j, x^j\rangle = 1\}$.

Proof:
Indeed, since $d^j \in \mathrm{int}\, K_j^*$, function ψ_j is well defined for all $u^j \in E_j$. Moreover,

$$\begin{aligned}
\min_\tau\{\, \tau : \; \tau d^j - u^j \in K_j^* \,\} &= \min_\tau \max_{x^j \in K_j} \{\, \tau + \langle u^j - \tau d^j, x^j\rangle \,\} \\
&= \max_{x^j \in K_j} \min_\tau\{\, \tau + \langle u^j - \tau d^j, x^j\rangle \,\} \\
&= \max_{x^j \in K_j} \{\, \langle u^j, x^j\rangle : \; \langle d^j, x^j\rangle = 1 \,\}.
\end{aligned}$$

Expression for $\partial\psi_j$ follows directly from representation (2.9). □

It is clear that $c^j - A_j^* y \in K_j^*$ if and only if $f_j(y) \stackrel{\mathrm{def}}{=} \psi_j(A_j^* y - c^j) \le 0$. However, before writing down the constraints of problem (2.7) in a functional form, we need to introduce an important normalization for their subgradients.

Namely, for each u^j, denote by $x^j(u^j) \in K_j$ an arbitrary optimal solution to problem (2.9). Then

$$f_j'(y) \;\stackrel{\mathrm{def}}{=}\; A_j\, x^j(A_j^* y - c^j) \;\in\; \partial f_j(y) \;\subset\; R^m. \tag{2.10}$$

We measure the size of this vector using the Hessian $\nabla^2 F_j^*(d^j)$ of a logarithmically homogeneous self-concordant barrier F_j^* for the dual cone K_j^*.

Lemma 2 *For any j, $1 \le j \le n$, we have*

$$\|f_j'(y)\|^2 \;\le\; \sigma_j^2 \;\stackrel{\mathrm{def}}{=}\; \lambda_{\max}\left(A_j \nabla^2 F_j^*(d^j) A_j^*\right). \tag{2.11}$$

Proof:
Denote $z^j = -\nabla F_j^*(d^j) \in \mathrm{int}\, K_j$, $x^j = x^j(A_j^* y - c^j)$. Then, $\nabla^2 F_j(z^j) = [\nabla^2 F_j^*(d^j)]^{-1}$ and

$$\|A_j\, x^j\|^2 \;=\; \|A_j[\nabla^2 F_*^j(d^j)]^{1/2} \cdot [\nabla^2 F_*^j(d^j)]^{-1/2} x^j\|^2 \;\le\; \sigma_j^2 \cdot \langle \nabla^2 F_j(z^j) x^j, x^j\rangle.$$

Hence, by Theorem of Recession Direction we have: $\langle \nabla^2 F_j(z^j) x^j, x^j\rangle \le \langle \nabla F_j(z^j), x^j\rangle^2 = \langle d^j, x^j\rangle^2 = 1$. □

Example 1 *1) If $K_j = R^1_+$, then $A_j = A e_j \in R^m$, where e_j is the corresponding coordinate vector in R^n. We can take $F_j(z) = -\ln z$ and $d^j = 1$. Then $\nabla^2 F_j(z^j) = 1$ and $\sigma_j^2 = \lambda_{\max}(A_j A_j^T) = \|A_j\|^2$.*

2) Let $K_j = S^p_+$, the cone of symmetric positive-semidefinite $p \times p$-matrices. Then we can take $F_j(z) = -\ln\det z$, and $z^j = d^j = I_p$, the unit $p \times p$-matrix. In this case, $\langle \nabla^2 F_j(z^j)h, h\rangle = \|h\|^2_F$. Note that the operator $A^*_j(y)$ is defined now as follows:

$$A^*_j(y) \;=\; \sum_{i=1}^m A^i_j y^i, \quad y \in R^m,$$

where A^i_j are symmetric $p \times p$-matrices. Thus, the scaling factor σ_j can be computed as

$$\sigma_j \;=\; \max_{\|y\|=1} \| \sum_{i=1}^m A^i_j y^i \|_F \;=\; \max_{\substack{\|y\|=1,\\ \|B\|_F=1}} \langle \sum_{i=1}^m A^i_j y^i, B\rangle \;=\; \max_{\|B\|_F=1} \left[\sum_{i=1}^m \langle A^i_j, B\rangle^2 \right]^{1/2}.$$

If $p \ll m$ and the operator A^*_j is sparse, then the last representation is preferable. □

We assume that all values σ_j, $j = 1, \dots, n$, are computed in advance, and they are available for the numerical scheme. Denote $g_j(y) = \frac{1}{\sigma_j} f_j(y)$. We come to the following representation of the dual problem (2.7):

$$\sup_{y \in R^m,\, s \in E^*} \left\{ \langle b, y\rangle : \; g(y) \stackrel{\rm def}{=} \max_{1 \le j \le n} g_j(y) \;\le\; 0 \right\}. \tag{2.12}$$

Denote by $j(y)$ the active index j such that $g_j(y) = g(y)$. Then

$$g'(y) \;=\; \tfrac{1}{\sigma_{j(y)}} A_{j(y)}\, x^{j(y)} \left(A^*_{j(y)} y - c^{j(y)} \right), \quad \|g'(y)\| \stackrel{(2.11)}{\le} 1. \tag{2.13}$$

Our new method generates the main minimization sequence in the dual space R^m, constructing at the same time an approximate primal solution. In both cases, the linear equality and inequality constraints can be violated. Our method has a single parameter, the step size $h > 0$. It always starts from $y_0 = 0$.

$$\begin{aligned} \textbf{For } k \ge 0 \textbf{ do:} \quad \textbf{if } g(y_k) \le h\textbf{, then (F):} \quad & y_{k+1} = y_k + h \cdot \tfrac{b}{\|b\|}, \\ \textbf{else (G):} \quad & y_{k+1} = y_k - g(y_k) \cdot g'(y_k). \end{aligned} \tag{2.14}$$

For $N \ge 0$, denote by $\mathcal{F}_N$ the set of iterations of type (F), which were used for generating the point y_{N+1}. And let $\mathcal{G}_N \stackrel{\rm def}{=} \{0, \dots, N\} \setminus \mathcal{F}_N$, $N_f \stackrel{\rm def}{=} |\mathcal{F}_N|$, and $N_g \stackrel{\rm def}{=} |\mathcal{G}_N|$. Thus, $N_f + N_g = N + 1$.

In view of definition of step (F), we have

$$c^j - A^*_j y_k + h\sigma_j d^j \;\in\; K^*_j, \; j = 1, \dots, n, \quad k \in \mathcal{F}_N. \tag{2.15}$$

In what follows, we denote by $\hat{d} \in K^*$ the vector with the components

$$\hat{d}^j \;=\; \sigma_j d^j, \quad j = 1, \dots, n. \tag{2.16}$$

Denote by $e_j(x^j) \in E$ the following vector:

$$e^i_j(x^j) \;=\; \begin{cases} x^j, & i = j, \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \dots, n.$$

Let us define the approximations for the optimal primal-dual solutions as follows:

$$\begin{array}{rcl} \bar{x}_N & \stackrel{\rm def}{=} & \frac{\|b\|}{hN_f} \sum\limits_{k \in \mathcal{G}_N} \frac{g(y_k)}{\sigma_{j(y_k)}} e_{j(y_k)} \left(x^{j(y_k)}(A^*_{j(y_k)} y_k - c^{j(y_k)}) \right) \in K, \\ \bar{y}_N & = & \frac{1}{N_f} \sum\limits_{k \in \mathcal{F}_N} y_k, \quad \bar{s}_N = c - A^T \bar{y}_N. \end{array} \tag{2.17}$$

This choice is motivated by the following relations:

$$\bar{s}^j_N = c^j - \frac{1}{N_f} \sum_{k \in \mathcal{F}_N} A^*_j y_k \stackrel{(2.15)}{\succeq_{K^*_j}} -h\sigma_j d^j, \tag{2.18}$$

$$\begin{array}{rcl} y_{N+1} & = & \frac{hN_f}{\|b\|} \cdot b - \sum\limits_{k \in \mathcal{G}_N} \frac{g(y_k)}{\sigma_{j(y_k)}} A e_{j(y_k)} \left(x^{j(y_k)}(A^*_{j(y_k)} y_k - c^{j(y_k)}) \right) \\ & \stackrel{(2.17)}{=} & \frac{hN_f}{\|b\|} \cdot (b - A\bar{x}_N). \end{array} \tag{2.19}$$

Let us study the performance of method (2.14).

Theorem 1 *Denote* $\hat{D} = 2\left(\frac{\langle \hat{d}, x^* \rangle}{\|b\|} + 1 \right)$. *For any* $N \geq 0$ *we have:*

$$N_f \geq \frac{1}{\hat{D}} \left(N + 1 - \frac{\|y^*\|^2}{h^2} \right). \tag{2.20}$$

If $N_f \geq 1$, *then*

$$\langle c, \bar{x}_N \rangle - \langle b, \bar{y}_N \rangle \leq \tfrac{1}{2} h \|b\|. \tag{2.21}$$

Finally, if

$$N + 1 > \frac{\|y^*\|^2}{h^2}, \tag{2.22}$$

then

$$\langle x^*, \bar{s}_N \rangle + \langle \bar{x}_N, s^* \rangle \leq h\|b\|, \tag{2.23}$$

and the residual in the primal-dual system vanishes as $N \to \infty$*:*

$$\frac{1}{\|b\|} \|b - A\bar{x}_N\| \leq \sqrt{\frac{\hat{D}}{N_f}} + \frac{\|y^*\|}{hN_f}. \tag{2.24}$$

Proof:
Denote $r_k = \|y_k - y^*\|$. For the sake of notation, denote $j_k = j(y_k)$, $x^{j_k}_k = x^{j_k}(A^*_{j_k} y_k - c^{j_k})$, and $e_{j_k} = e_{j_k}(x^{j_k}_k)$.

If $k \in \mathcal{G}_N$, then

$$\begin{array}{rcl} r^2_{k+1} & = & r^2_k - 2g(y_k)\langle g'(y_k), y_k - y^* \rangle + g^2(y_k)\|g'(y_k)\|^2 \\ & \stackrel{(2.13)}{\leq} & r^2_k - 2\frac{g(y_k)}{\sigma_{j_k}} \langle A_{j_k} x^{j_k}_k, y_k - y^* \rangle + g^2(y_k). \end{array}$$

Note that

$$\begin{array}{rcl} \frac{1}{\sigma_{j_k}} \langle A_{j_k} x^{j_k}_k, y_k - y^* \rangle & = & \frac{1}{\sigma_{j_k}} \langle x^{j_k}_k, A^*_{j_k} y_k - c^{j_k} \rangle - \frac{1}{\sigma_{j_k}} \langle x^{j_k}_k, A^*_{j_k} y^* - c^{j_k} \rangle \\ & = & g(y_k) + \frac{1}{\sigma_{j_k}} \langle x^{j_k}_k, c^{j_k} - A^*_{j_k} y^* \rangle \geq g(y_k). \end{array}$$

Therefore,

$$r_{k+1}^2 - r_k^2 \;\leq\; -g^2(y_k) - 2\langle \tfrac{g(y_k)}{\sigma_{j_k}} e_{j_k}, s^* \rangle.$$

Thus, in this case $r_{k+1}^2 \leq r_k^2 - h^2$. Hence, condition (2.22) implies $N_f \geq 1$. Further, if $k \in \mathcal{F}_N$, then

$$r_{k+1}^2 \;=\; r_k^2 + \tfrac{2h}{\|b\|}\langle b, y_k - y^* \rangle + h^2.$$

Now, summing up all these equalities for $k = 0, \ldots, N$, we obtain:

$$r_{N+1}^2 \;\leq\; r_0^2 - h^2 N_g - 2\tfrac{hN_f}{\|b\|}\langle \bar{x}_N, s^* \rangle + 2\tfrac{hN_f}{\|b\|}\langle b, \bar{y}_N - y^* \rangle + h^2 N_f. \tag{2.25}$$

This relation leads to several important consequences.

1. In view of (2.19), inequality (2.25) can be written as follows:

$$\left\| \tfrac{hN_f}{\|b\|} \cdot (b - A\bar{x}_N) - y^* \right\|^2 \leq \|y^*\|^2 + h^2(N_f - N_g) - 2\tfrac{hN_f}{\|b\|}[\langle b, y^* - \bar{y}_N \rangle + \langle \bar{x}_N, s^* \rangle].$$

Note that $\langle b, y^* - \bar{y}_N \rangle + \langle \bar{x}_N, s^* \rangle = \langle c, \bar{x}_N \rangle - \langle b, \bar{y}_N \rangle + \langle y^*, b - A\bar{x}_N \rangle$. Hence, the above inequality can be rewritten in the following form:

$$0 \;\leq\; \tfrac{h^2 N_f^2}{\|b\|^2} \cdot \|b - A\bar{x}_N\|^2 \;\leq\; h^2(N_f - N_g) - 2\tfrac{hN_f}{\|b\|}[\langle c, \bar{x}_N \rangle - \langle b, \bar{y}_N \rangle].$$

Since $N_f \geq 1$, we obtain (2.21).

2. Since $\langle s^*, x^* \rangle = 0$, we have

$$\langle b, y^* \rangle - \langle b, \bar{y}_N \rangle \;=\; \langle Ax^*, y^* - \bar{y}_N \rangle \;=\; \langle x^*, \bar{s}_N - s^* \rangle \;=\; \langle x^*, \bar{s}_N \rangle.$$

Thus, we get

$$2\tfrac{hN_f}{\|b\|}\left[\langle x^*, \bar{s}_N \rangle + \langle \bar{x}_N, s^* \rangle\right] \overset{(2.25)}{\leq} r_0^2 + 2h^2 N_f - h^2(N+1). \tag{2.26}$$

Assuming now relation (2.22), we obtain the estimate (2.23) for the primal-dual gap.

3. Finally, let us prove that the residual in the primal system of linear equations vanishes as $N \to \infty$. Indeed,

$$N_f \langle b, \bar{y}_N \rangle \;=\; \sum_{k \in \mathcal{F}_N} \langle x^*, A^T y_k \rangle \overset{(2.15)}{\leq} \sum_{k \in \mathcal{F}_N} \langle x^*, c + h\hat{d} \rangle \;=\; N_f[f^* + h\langle \hat{d}, x^* \rangle].$$

Therefore, from (2.25) we get

$$\begin{aligned} r_{N+1}^2 \;&\leq\; r_0^2 + 2\tfrac{h^2 N_f}{\|b\|}\langle \hat{d}, x^* \rangle + h^2\,(N_f - N_g) \;=\; r_0^2 + h^2(N_f(\hat{D} - 1) - N_g) \\ &=\; r_0^2 + h^2(N_f \hat{D} - N - 1). \end{aligned}$$

This inequality has two consequences. The first one is (2.20). For the second one, assuming that (2.22) is true, we obtain

$$h\sqrt{N_f \hat{D}} \;\geq \left\| \tfrac{hN_f}{\|b\|}(b - A\bar{x}_N) - y^* \right\| \;\geq\; \tfrac{hN_f}{\|b\|}\|b - A\bar{x}_N\| - \|y^*\|.$$

Therefore, (2.24) follows. □

We are ready to justify an implementable version of method (2.14) with explicit choice of parameters. For the sake of simplicity, we assume that $c \in \operatorname{int} K^*$. Thus, $N_f \geq 1$ for all $N \geq 0$.

Our method has three positive accuracy parameters, ϵ_f, ϵ_g, and ϵ_a. Its goal is to generate an approximate primal-dual solution $(\hat{x}, \hat{y}, \hat{s})$ of the problem (2.3), (2.4) satisfying the following conditions:

$$\begin{aligned} \hat{x} \in K, \quad \hat{s} &= c - A^T \hat{y}, \quad \hat{s} + \epsilon_g \cdot d \in K^*, \\ \langle c, \hat{x} \rangle - \langle b, \hat{y} \rangle &\leq \epsilon_f, \quad \|A\hat{x} - b\| \leq \epsilon_a. \end{aligned} \tag{2.27}$$

In accordance to (2.18) and (2.21), we need to choose

$$h = \min \left\{ \epsilon_f \cdot \tfrac{2}{\|b\|}, \; \epsilon_g \cdot \tfrac{1}{\max\limits_{1 \leq j \leq n} \sigma_j} \right\}. \tag{2.28}$$

It remains to introduce in (2.14) the following stopping criterion:

$$\|A\bar{x}_N - b\| \leq \epsilon_a. \tag{2.29}$$

Note that all other conditions in (2.27) are satisfied automatically, by the choice (2.28) of parameter h. It remains to obtain the complexity bound for reaching the goal (2.29).

In view of (2.24), it will be satisfied for

$$N_f \geq \max \left\{ \tfrac{2r_0\|b\|}{h\epsilon_a}, \tfrac{4\|b\|^2 \hat{D}}{\epsilon_a^2} \right\}.$$

In accordance to (2.20), we come to the following estimate:

$$N + 1 \geq \left(\tfrac{r_0}{h}\right)^2 + \hat{D} \cdot \max \left\{ \tfrac{2r_0\|b\|}{h\epsilon_a}, \tfrac{4\|b\|^2 \hat{D}}{\epsilon_a^2} \right\}. \tag{2.30}$$

Thus, assuming that all accuracies in (2.27) are of the same order ϵ, we get $O(\frac{1}{\epsilon^2})$ bound for our scheme. However, note that this pessimistic bound is not incorporated directly in the method. Indeed, the stopping criterion (2.29) can be satisfied much earlier.

3 Solving Huge-Scale Conic Problems

Let us show that method (2.14), (2.29) can be used for solving huge-scale optimization problems. We need to be sure that all its operations take into account the sparsity of initial data. In this section, $p(x)$ denotes the number of nonzero elements of vector x, and $\sigma(x)$ is the set of corresponding indexes.

Let us estimate the computational cost of our method for the standard Linear Programming problem with $K = R^n_+$. As it was explained in Example 1, in this case $\sigma_j = \|Ae_j\|$, $j = 1, \ldots, n$.

In what follows, we assume that the data of problem (2.3) is uniformly sparse:

$$\begin{array}{rcl} p(c) & \leq & r, \quad p(A^T e_i) \leq r, \; i = 1, \ldots, m, \\ p(b) & \leq & q, \quad p(Ae_j) \leq q, \; j = 1, \ldots, n, \end{array} \tag{3.1}$$

with some $r \ll n$ and $q \ll m$. These conditions are satisfied, for example, by a matrix with a few nonzero diagonals.

Denote $s_k = c - A^T y_k$. Before we start the iterative procedure, it is necessary to perform the following preliminary computations:

- Compute the norm $\|b\|$, ($O(q)$ a.o.).
- Compute the values σ_j, $j = 1, \ldots, n$, ($O(p(A))$ a.o.).
- Set $y_0 = 0$ and $s_0 \stackrel{\text{def}}{=} c$, ($m + n$ a.o.).
- For computing the value $g(y_0)$, fill the Binary Tree Table by vector s_0, (n operations).
- Choose the step size h in accordance to (2.28), ($O(n)$ a.o.).

Clearly, all these computations need $O(p(A))$ operations.

Let us look now at the complexity of one iteration of method (2.14), (2.29). We assume that at the beginning of each iteration the value $g(y_k)$ is already computed. Hence, we know $j(y_k)$.

At each iteration of method (2.14) our main operation is the update of vector y_k by a sparse direction δ_k: $y_{k+1} = y_k + \delta_k$. This direction can be either $\delta_k = h \cdot \frac{b}{\|b\|}$, or $\delta_k = -\frac{g(y_k)}{\sigma_{j(y_k)}} \cdot Ae_{j(y_k)}$. In both situations, $p(\delta_k) \stackrel{(3.1)}{\leq} q$. Therefore this update takes $O(q)$ operations at most. After that, we need to compute the new slack variables s_{k+1}. As it was suggested in [1], it is better to do this in parallel with updating the value $g(y_{k+1})$. Namely, we start with $s_+ = s_k$, and perform the following operations:

For $j \in \sigma(\delta_k)$**,** $i \in \sigma(A^T e_j)$ **iterate:**

1. Update a single entry of slack variables: $s_+^i = s_+^i - A_{i,j}\delta_k^j$. (3.2)

2. Update $\max\limits_{1 \leq l \leq n} s_+^l$ by the Binary Tree Table (see Section 3 in [1]).

In the end, we accept s_+ as s_{k+1} and $g(y_{k+1}) = \max\limits_{1 \leq l \leq n} s_+^l$. If for all vectors s_k are stored in the same array of computer memory, then the complexity of the full loop (3.2) does not exceed $O(rq \log_2 n)$ operations (see (3.1)).

Further, in accordance to (2.19), our stopping criterion will be

$$\|y_{k+1}\|^2 \quad \leq \quad \left[\epsilon_a \cdot \frac{hk_f}{\|b\|}\right]^2. \tag{3.3}$$

Therefore, at each iteration it is necessary to update the value $\|y_{k+1}\|^2$. Since

$$\|y_{k+1}\|^2 \quad = \quad \|y_k\|^2 + 2\langle y_k, \delta_k \rangle + \|\delta_k\|^2,$$

this computation takes $O(q)$ operations at most. Thus, we conclude that the computational cost of one iteration of method (2.14), (2.29) does not exceed

$$O(rq \log_2 n) \tag{3.4}$$

operations.

It remains to discuss the rules for generating the output. During the main optimization process, it is easy to update the vector $\hat{x}_k = \sum\limits_{i \in \mathcal{G}_k} \frac{g(y_i)}{\sigma_{j(y_i)}} e_{j(y_i)} \in R^n_+$. Then in the end of the process we can deliver $\bar{x}_N \stackrel{(2.17)}{=} \frac{\|b\|}{hN_f} \hat{x}_N$.

The rules for generated acceptable dual solutions are more complicated. Indeed, we cannot use definition (2.17) for generating $\bar{y}_N$ and $\bar{s}_N$ since they may result in $O(m+n)$ operations at each iteration. Therefore let us apply the *Double Run* strategy suggested in [1]. Namely, let us deliver a vector of dual variables with the best value of the objective function. For doing that, define

$$k^*_N \;\; = \;\; \arg \max_{0 \le k \le N} \langle b, y_k \rangle.$$

This number can be easily updated during the optimization process (we need also to update the values $\langle b, y_k \rangle$). After termination of the main process by stopping criterion (3.3), we run it again and stop earlier at the moment k^*_N. Then the current state of the dual variables and the slack vector will correspond to $y_{k^*_N}$ and $s_{k^*_N}$. Clearly, these points satisfy the performance guarantees (2.21) and (2.23). With these rules for generating the output, the total complexity of our method remains on the level $O\left(\frac{rq}{\epsilon^2} \log_2 n\right)$ operations.

4 Application: Truss Topology Design

Truss Topology Design problem consists in finding the best mechanical structure resisting to an external force with an upper bound for the total weight of construction. Its mathematical formulation looks as follows:

$$\min_{w \in R^N_+} \{ \langle \bar{f}, u \rangle : \; A(w)u = \bar{f}, \langle e, w \rangle = T \}, \tag{4.1}$$

where $\bar{f}$ is a vector of external forces, $u \in R^{2n}$ is a vector of virtual displacements of n nodes in R^2, w is a vector of N bars, and T is the total weight of construction. The compliance matrix $A(w)$ has the following form:

$$A(w) \;\; = \;\; \sum_{i=1}^{N} w_i a_i a_i^T,$$

where $a_i \in R^{2n}$ are the vectors describing the interactions of two nodes connected by an arc. These vectors are very sparse: for 2D-model they have at most 4 nonzero elements.

Let us rewrite the problem (4.1) as a Linear Programming problem.

$$\begin{aligned}
& \min_{u,w}\{\langle \bar f, u\rangle : A(w)u = \bar f,\ w \ge 0,\ \langle e, w\rangle = T\} \\
= \ & \min_{w}\{\langle \bar f, A^{-1}(w)\bar f\rangle :\ w \in \Delta(T) \stackrel{\rm def}{=} \{w \ge 0,\ \langle e, w\rangle = T\}\} \\
= \ & \min_{w\in\Delta(T)} \max_{u}\{2\langle \bar f, u\rangle - \langle A(w)u, u\rangle\} \ \ge\ \max_{u} \min_{w\in\Delta(T)} \{2\langle \bar f, u\rangle - \langle A(w)u, u\rangle\} \qquad (4.2) \\
= \ & \max_{u}\{2\langle \bar f, u\rangle - T \max_{1\le i\le N}\langle a_i, u\rangle^2\} \ =\ \max_{\lambda, y}\{2\lambda\langle \bar f, y\rangle - \lambda^2 T \max_{1\le i\le N}\langle a_i, y\rangle^2\} \\
= \ & \max_{y} \tfrac{\langle \bar f, y\rangle^2}{T \max\limits_{1\le i\le N}\langle a_i, y\rangle^2} \ =\ \tfrac{1}{T}\left(\max_{y}\{\langle \bar f, y\rangle :\ \max_{1\le i\le N}|\langle a_i, y\rangle| \le 1\}\right)^2.
\end{aligned}$$

Note that for the inequality in the third line we do not need any assumption.

Denote by y^* the optimal solution of the optimization problem in the brackets. Then there exist multipliers $x^* \in R^N_+$ such that

$$\bar f \ = \ \sum_{i\in J_+} a_i x_i^* - \sum_{i\in J_-} a_i x_i^*, \quad x_i^* \ =\ 0,\ i \notin J_+ \bigcap J_-, \qquad (4.3)$$

where $J_+ = \{i : \langle a_i, y^*\rangle = 1\}$, and $J_- = \{i : \langle a_i, y^*\rangle = -1\}$. Multiplying the first equation in (4.3) by y^*, we get

$$\langle \bar f, y^*\rangle \ = \ \langle e, x^*\rangle. \qquad (4.4)$$

Note that the first equation in (4.3) can be written as

$$\bar f \ = \ A(x^*)y^*. \qquad (4.5)$$

Let us reconstruct now the solution of the primal problem. Denote

$$w^* \ = \ \tfrac{T}{\langle e, x^*\rangle}\cdot x^*, \quad u^* \ =\ \tfrac{\langle e, x^*\rangle}{T}\cdot y^*. \qquad (4.6)$$

Then, in view of (4.5) we have $\bar f = A(w^*)u^*$, and $w^* \in \Delta(T)$. Thus, the pair (4.6) is feasible for the primal problem. On the other hand,

$$\langle \bar f, u^*\rangle \ = \ \langle \bar f, \tfrac{\langle e, x^*\rangle}{T}\cdot y^*\rangle \ =\ \tfrac{1}{T}\cdot\langle e, x^*\rangle\cdot\langle \bar f, y^*\rangle \ \stackrel{(4.4)}{=}\ \tfrac{1}{T}\cdot\langle \bar f, y^*\rangle^2.$$

Thus, the duality gap in the chain (4.2) is zero, and the pair (w^*, u^*), defined by (4.6) is the optimal solution of the primal problem.

The above discussion allows us to concentrate on the following (dual) Linear Programming problem:

$$\max_{y}\{\langle \bar f, y\rangle :\ \max_{1\le i\le N}\langle \pm a_i, y\rangle \le 1\}, \qquad (4.7)$$

which we are going to solve by method (2.14). For our experiments, we choose a *local* truss: each node is connected only with eight neighbors. This allows to apply the technique described in Section 3. As a result, in our experiments the computational cost of each iteration grows as $O(\log_2 N)$. The whole grid was formed by square cells, K cells in each

row, with K rows. Thus, the total number of nodes was $(K+1)^2$. $K+1$ nodes of the left vertical border were fixed. Thus, the total number of nodes was $K(K+1)$.

In the computational results presented below we did not bound the number of iterations. Our main parameter was the accuracy $\epsilon = 0.02$, which we used in the stopping criterion $\|Ax - \bar{f}\| \leq \epsilon$.

K	$N/10^3$	Time (sec) for 10^6 it.	$\|y_{N+1}\|$	# of Iterations div. by / 10^6	$N_f/10^6$	Time (sec)
1	0.01	0.9	4	0.07	0.02	0.06
2	0.02	3.8	5	0.17	0.04	0.6
4	0.07	6.9	8	0.5	0.06	3.5
8	0.26	9.6	14	1.8	0.1	17
16	1	11.9	27	6.8	0.2	81
32	4	13.4	61	28	0.5	379
64	17	16.1	153	121	1.2	1958
128	66	17.1	280	333	2	5695
256	262	18.4	347	488	2.6	8993
512	1049	20.2	372	561	2.8	11335
1024	4195	22.4	347	524	2.6	11726

We can see that our method managed to solve all problems. The largest problem in the table has eight million inequality constraints. In the third column we present the computational time spent for 10^6 iterations of the method. Clearly, it grows logarithmically with the dimension of the problem. The most dangerous characteristic of the problem is the size of the dual solution. The total number of iterations is proportional to the square of this value. As we can see, for this type of problems the size is finally stabilized on a reasonable level. Consequently, the total computational time for the last three problems is not too much different. In our opinion, the last line shows that our approach can be indeed very efficient for solving sparse optimization problems of very big size. Note that all our experiments were performed on a standard PC.

References

[1] Yu. Nesterov. Subgradient methods for huge-scale optimization problems. CORE Discussion Paper 2012/2 (January 2012).

[2] Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. CORE Discussion Paper 2010/2, (January 2010).

[3] Yu. Nesterov, A. Nemirovskii. *Interior point polynomial methods in convex programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[4] P. Richtrik, M. Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Edinburgh University, April 2011 (submitted to *Mathematical Programming*).

[5] P. Tseng, S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, **117**(1-2), 387-423, (2009).