A RANDOM COORDINATE DESCENT ALGORITHM FOR SINGLY LINEAR CONSTRAINED SMOOTH OPTIMIZATION*

I. NECOARA† AND A. PATRASCU‡

Abstract. In this paper we develop a novel randomized block-coordinate descent method for minimizing multi-agent convex optimization problems with singly linear coupled constraints over networks and prove that it obtains in expectation an ϵ accurate solution in at most $\mathcal{O}(\frac{1}{\lambda_2(Q)\epsilon})$ iterations, where $\lambda_2(Q)$ is the second smallest eigenvalue of a matrix Q that is defined in terms of the probabilities and the number of blocks. However, the computational complexity per iteration of our method is much simpler than of a method based on full gradient information and each iteration can be computed in a completely distributed way. We focus on how to choose the probabilities to make this randomized algorithm to converge as fast as possible and we arrive at solving a sparse SDP. Numerical tests confirm that on huge optimization problems our method is much more numerically efficient than methods based on full gradient.

Key words. coordinate descent method, randomized algorithm, large-scale convex optimization,support vector machine, compressed sensing

AMS subject classifications. 90C06, 90C25, 90C52

1. Introduction.

In many application fields, the notion of networks has emerged as a central, unifying concept for solving different problems in control theory, economics and computer science. The goal of this paper is to develop an efficient (block) coordinate descent type algorithm for solving singly linear constrained optimization problems that appear in the framework of networks. The main features of the problem that we consider in this paper are *its huge dimension* and *the incomplete structure of information*, both being an obstacle for total gradient computations (these features usually appear in distributed systems and control). An appropriate way to approach these problems is then through coordinate descent methods [1]. These methods have received increasing attention in the last years due to recent results in support vector machine [2], compressed sensing [3], protein loop closure [4] and optimization [5].

The main differences in all variants of coordinate descent methods consist in the criterion of choosing at each iteration the coordinate over which we minimize. Two classical criteria used often in algorithms are the cyclic and the maximal descent coordinate [1]. Another interesting approach is based on random coordinate descent. Recent complexity results on random coordinate descent methods were obtained by Nesterov in [7] and the extension to composite objective functions case was given in [9]. However, both papers studied optimization models where the constraint set is decoupled (i.e. characterized by cartesian product).

In this paper we provide a random coordinate descent method suited for large scale problems where the information cannot be gather centrally, but rather the information is distributed over the network. Moreover, we focus here on singly linear constrained optimization problems (i.e. the constraints are coupled) and we prove for our method

*The research leading to these results has received funding from: the European Union, FP7/2007–2013 under grant agreement no 248940; CNCSIS-UEFISCSU (project TE, no. 19/11.08.2010); ANCS (project PN II, no. 80EU/2010); Sectoral Operational Programme Human Resources Development 2007-2013 through the Financial Agreements POSDRU/89/1.5/S/62557.

†Automation and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania (ion.necoara@acse.pub.ro).

‡Automation and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania (andrei.patrascu@acse.pub.ro).

that it obtains in expectation an ϵ accurate solution in at most $\mathcal{O}(\frac{1}{\lambda_2(Q)\epsilon})$ iterations, where $\lambda_2(Q)$ is the second smallest eigenvalue of a matrix Q that is defined in terms of the probabilities and the number of blocks. Moreover, the complexity per iteration of our method can be much simpler than of a method based on full gradient information.

This paper is organized as follows. In Section 2 the problem formulation, assumptions and notations which we will use for the remainder of the paper are presented. Then, in Section 3 we derive a randomized block coordinate descent method for which we prove the rate of convergence in expectation. We also focus on how to choose the probabilities to make this randomized algorithm to converge as fast as possible and we arrive at solving a sparse SDP. Finally, we present some numerical results for our method that show its efficiency on huge sparse problems.

2. Problem formulation. In this paper we develop a random coordinate descent method for singly linear constrained convex minimization problems of the following form:

$$f^* = \min_{x \in \mathbb{R}^n} \{f(x) : \quad \text{s.t. } a^T x = 0\}, \tag{2.1}$$

where the objective function f is smooth and convex, and $a \in \mathbb{R}^n$. Singly linear constrained optimization problems arise in many areas such as resource allocation in economic systems [10] or distributed computer systems [11]. In [12] a distributed weighted gradient method was proposed to solve a similar problem as in (2.1), in particular the authors consider strongly convex function f. For our problem (2.1) we associate a network composed of several nodes $V = \{1, 2, \cdots, N\}$, e.g. internet users, subsystems, etc., that can exchange information according to a communication graph

$$G = (V, E),$$

where E denotes the set of edges, i.e. $(i, j) \in E \subseteq V \times V$ models that node j sends information to node i. We assume that the graph G is undirected and connected.

We work in the space $\mathbb{R}^n$ composed by column vectors. For $x, y \in \mathbb{R}^n$ denote the standard Euclidian inner product $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ and $\|\cdot\|$ denotes the Euclidian norm. We use the same notation $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ for spaces of different dimension. For convenience, sometimes we also use $x^T y = \sum_{i=1}^n x_i y_i$ (especially when we work with matrices). We denote the feasible set of problem (2.1) by:

$$S = \{x \in \mathbb{R}^n | \ a^T x = 0\}.$$

Let us consider a decomposition of the dimension of the variables: $n = \sum_{i=1}^N n_i$. Let us divide identity matrix into:

$$I_n = \begin{bmatrix} U_1 & \cdots & U_N \end{bmatrix}, \ U_i \in \mathbb{R}^{n \times n_i}$$

and use the following notation

$$\nabla_i f(x) = U_i^T \nabla f(x).$$

Hence, for any $x = [x_1 \cdots x_N]^T \in \mathbb{R}^n$ we can write $x = \sum_{i=1}^N U_i x_i$.

We notice that the KKT conditions of problem (2.1) are: $x^* = \begin{bmatrix} x_1^{*T} \cdots x_N^{*T} \end{bmatrix}^T \in \mathbb{R}^n$ is an optimal point for convex problem (2.1) if and only if there exists some scalar $\lambda^* \in \mathbb{R}$ such that:

$$a^T x^* = 0, \quad \nabla f(x^*) = \lambda^* a. \tag{2.2}$$

ASSUMPTION 1. *We assume that function f has a component-wise Lipschitz continuous gradient with constants L_i, i.e.:*

$$\|\nabla_i f(x + U_i h_i) - \nabla_i f(x)\| \leq L_i \|h_i\| \quad \forall\, x \in \mathbb{R}^n,\ h_i \in \mathbb{R}^{n_i},\ i \in V.$$

3. A Random Coordinate Descent Method. In this section we present a distributed algorithm to solve (2.1), where only neighbors are required to communicate with each other. We first show that if the function f has component-wise Lipschitz continuous gradient, then it has in every pair (i, j) Lipschitz continuous gradient.

LEMMA 3.1. *[6] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth convex function satisfying Assumption 1, then for each pair (i, j) with $i \neq j$, denoting $L_{ij} = L_i + L_j$, we have:*

$$\left\| \begin{bmatrix} \nabla_i f(x + U_i s_i + U_j s_j) - \nabla_i f(x) \\ \nabla_j f(x + U_i s_i + U_j s_j) - \nabla_j f(x) \end{bmatrix} \right\| \leq L_{ij} \left\| \begin{bmatrix} s_i \\ s_j \end{bmatrix} \right\| \quad \forall x \in \mathbb{R}^n,\ s_i \in \mathbb{R}^{n_i},\ s_j \in \mathbb{R}^{n_j}.$$

As a consequence of the above lemma we have [6, 8]:

$$f(x + U_i s_i + U_j s_j) \leq f(x) + \langle \begin{bmatrix} \nabla_i f(x) \\ \nabla_j f(x) \end{bmatrix}, \begin{bmatrix} s_i \\ s_j \end{bmatrix} \rangle + \frac{L_{ij}}{2} \left\| \begin{bmatrix} s_i \\ s_j \end{bmatrix} \right\|^2. \tag{3.1}$$

Given $x = \begin{bmatrix} x_1^T \cdots x_N^T \end{bmatrix}^T$ in the feasible set S, we choose the coordinate pair (i, j) with probability p_{ij}. Let the next feasible iterate be defined as:

$$x^+ = x + U_i d_i + U_j d_j,$$

where the pair of directions (d_i, d_j) are given by:

$$(d_i, d_j) = \arg \min_{s_i, s_j:\ a_i^T s_i + a_j^T s_j = 0} \langle \begin{bmatrix} \nabla_i f(x) \\ \nabla_j f(x) \end{bmatrix}, \begin{bmatrix} s_i \\ s_j \end{bmatrix} \rangle + \frac{L_{ij}}{2} \left\| \begin{bmatrix} s_i \\ s_j \end{bmatrix} \right\|^2. \tag{3.2}$$

We can observe that the optimal solution of (3.2) can be computed analytically:

$$(d_i, d_j) = -\frac{1}{L_{ij}} \left(\nabla_{ij} f(x) - \frac{a_{ij} a_{ij}^T}{a_{ij}^T a_{ij}} \nabla_{ij} f(x) \right),$$

where $\nabla_{ij} f(x) = \begin{bmatrix} \nabla_i f(x) \\ \nabla_j f(x) \end{bmatrix}$ and $a_{ij} = \begin{bmatrix} a_i \\ a_j \end{bmatrix}$. Now, if the starting point x_0 is feasible and assume some probability distribution $(p_{ij})_{(i,j) \in E}$ available over the undirected graph G, we can present our random coordinate descent method:

Random Coordinate Descent (RCD) Method

0) Choose randomly the pair (i_k, j_k). Set $x_{k+1}^l = x_k^l, \forall l \notin \{i_k, j_k\}$

1) $x_{k+1}^{i_k} = x_k^{i_k} - \dfrac{1}{L_{i_k j_k}} \left(\nabla_{i_k} f(x_k) - \dfrac{a_{i_k} a_{i_k j_k}^T}{a_{i_k j_k}^T a_{i_k j_k}} \nabla_{i_k j_k} f(x_k) \right)$

2) $x_{k+1}^{j_k} = x_k^{j_k} - \dfrac{1}{L_{i_k j_k}} \left(\nabla_{j_k} f(x_k) - \dfrac{a_{j_k} a_{i_k j_k}^T}{a_{i_k j_k}^T a_{i_k j_k}} \nabla_{i_k j_k} f(x_k) \right).$

The algorithm (RCD) updates at each iteration k only two components of x_k, so that numerical complexity per iteration for sparse problems is very cheap compared to full gradient methods. Moreover, in our algorithm we maintain feasibility at each iteration. In the sequel we use the random variable $\omega_k = \{(i_0, j_0), \ldots, (i_k, j_k)\}$.

3.1. Rate of convergence in expectation. In this section we derive the expected rate of convergence for our algorithm (RCD). Relation (3.1) give us the following decrease in objective function f:

$$f(x^+) \leq f(x) - \frac{1}{2L_{ij}} \nabla_{ij} f(x)^T \left(I_{n_i+n_j} - \frac{a_{ij} a_{ij}^T}{a_{ij}^T a_{ij}} \right) \nabla_{ij} f(x). \tag{3.3}$$

We denote by $Q_{ij} \in \mathbb{R}^{n \times n}$ a symmetric matrix with all blocks zero except:

$$Q_{ij}^{ii} = I_{n_i} - \frac{a_i a_i^T}{a_{ij}^T a_{ij}}, \;\; Q_{ij}^{ij} = -\frac{a_i a_j^T}{a_{ij}^T a_{ji}}, \;\; Q_{ij}^{jj} = I_{n_j} - \frac{a_j a_j^T}{a_{ij}^T a_{ij}}.$$

It is straightforward to see that $Q_{ij} a = 0$ for all pairs (i,j) with $i \neq j$ and Q_{ij} is also positive semidefinite (notation $Q_{ij} \succeq 0$). Let us define the matrix:

$$Q = \sum_{(i,j) \in E} \frac{p_{ij}}{L_{ij}} Q_{ij},$$

that is also symmetric and positive semidefinite, since $\frac{p_{ij}}{L_{ij}} > 0$ for all $(i,j) \in E$.

LEMMA 3.2. *[6] If the graph G is connected, then $\lambda_2(Q) > 0$.*

Taking the expectation in (3.3) over the random pair (i,j) we get:

$$E_{ij}[f(x^+)] \leq f(x) - \frac{1}{2} \nabla f(x)^T Q \nabla f(x). \tag{3.4}$$

We now introduce the following distance:

$$R(x_0) = \max_{x: f(x) \leq f(x_0)} \max_{x^* \in X^*} \|x - x^*\|,$$

which measures the size of the level set of f given by x_0. Moreover, for a given vector $w \in \mathbb{R}^n$, we denote with $w_\perp$ the projection of this vector w onto the subspace S that is orthogonal to the vector a. It follows immediately that $w_\perp$ is given by:

$$w_\perp = (I_n - \frac{a a^T}{a^T a}) w.$$

Let us denote

$$\phi_k = E_{\omega_k}[f(x_k)].$$

Then, we can derive the convergence rate of the method (RCD):

THEOREM 3.3. *Let f be convex function satisfying Assumption 1. The algorithm (RCD) generates a sequence x_k with the following expected rate of convergence:*

$$\phi_k - f^* \leq \frac{R^2(x_0)}{\lambda_2(Q) k}. \tag{3.5}$$

Proof. From (3.4) we have that:

$$E_{i_k j_k}[f(x_{k+1})] \leq f(x_k) - \frac{1}{2} \nabla f(x_k)^T Q \nabla f(x_k).$$

Note that $\nabla f(x_k)$ can be written as $\nabla f(x_k) = \alpha a + (\nabla f(x_k))_\perp$, for some scalar α. Since $Qa = 0$, we have that previous inequality does not change if we replace $\nabla f(x_k)$ with $(\nabla f(x_k))_\perp$:

$$E_{i_k j_k}[f(x_{k+1})] \leq f(x_k) - \frac{1}{2} (\nabla f(x_k))_\perp^T Q (\nabla f(x_k))_\perp .$$

The previous inequality restricted to the orthogonal complement of the span of vector a can be bounded as:

$$E_{i_k j_k}[f(x_{k+1})] \leq f(x_k) - \frac{1}{2}\lambda_2(Q) \| (\nabla f(x_k))_\perp \|^2. \tag{3.6}$$

If $(\nabla f(x_k))_\perp = 0$, which is equivalent to $\nabla f(x_k) = \beta a$ for some scalar β, then the KKT conditions (2.2) hold at x_k, i.e. x_k is optimal for optimization problem (2.1). We conclude that $E_{i_k j_k}[f(x_{k+1})] < f(x_k)$ provided that $(\nabla f(x_k))_\perp \neq 0$.

Since x_k and x^* are feasible, i.e. $a^T x_k = 0$ and $a^T x^* = 0$, we have that

$$\langle \nabla f(x_k), x^* - x_k \rangle = \langle \nabla f(x_k) - \frac{a^T \nabla f(x_k)}{a^T a} a, x^* - x_k \rangle.$$

Using that $(\nabla f(x_k))_\perp = \nabla f(x_k) - \frac{a a^T}{a^T a} \nabla f(x_k)$, then from the convexity of f we have the following relations:

$$\begin{aligned} f^* &\geq f(x_k) + \langle \nabla f(x_k), x^* - x_k \rangle = f(x_k) + \left\langle \nabla f(x_k) - \frac{a^T \nabla f(x_k)}{a^T a} a, x^* - x_k \right\rangle \\ &\geq f(x_k) + \langle (\nabla f(x_k))_\perp , x^* - x_k \rangle \end{aligned}$$

and then using Cauchy-Schwartz inequality we arrive at:

$$f(x_k) - f^* \leq \langle (\nabla f(x_k))_\perp , x_k - x^* \rangle \leq \| (\nabla f(x_k))_\perp \| \| x_k - x^* \| \leq R(x_0) \| (\nabla f(x_k))_\perp \|.$$

From (3.6) and the above inequality we get:

$$E_{i_k j_k}[f(x_{k+1}) - f^*] \leq f(x_k) - f^* - \frac{\lambda_2(Q)}{2R^2(x_0)} (f(x_k) - f^*)^2 . \tag{3.7}$$

Taking expectation in both sides of this inequality in ω_{k-1} and by denoting $\Delta_k = \phi_k - f^*$ we obtain the following inequality: $\Delta_{k+1} \leq \Delta_k - \frac{\lambda_2(Q)}{2R(x_0)^2} \Delta_k^2$. Using arguments as in the convergence of the full gradient method [8] we obtain (3.5). □

3.2. Selections for probabilities. We have several choices for the probabilities p_{ij}, which our randomized block coordinate descent algorithm (RCD) depends on. For example, we can choose uniform probabilities in order to determine the selection of the pair $(i, j) \in E$ at each iteration of algorithm (RCD), i.e.

$$p_{ij}^0 = \frac{1}{\sharp E}, \tag{3.8}$$

where $\sharp E$ denotes the cardinality of the set of edges E in the graph G. Another choice is to choose the probabilities dependent on the Lipschitz constants L_{ij}:

$$p_{ij}^\alpha = \frac{L_{ij}^\alpha}{L^\alpha}, \text{ where } L^\alpha = \sum_{(i,j)\in E} L_{ij}^\alpha, \ \alpha \geq 0. \tag{3.9}$$

Finally, from the convergence rate of our method $\phi_k - f^* \leq \frac{R^2(x_0)}{\lambda_2(Q)k}$, it follows that we can choose the matrix Q such that $\lambda_2(Q)$ is as large as possible, i.e:

$$\max_{Q \in \mathcal{M}} \lambda_2(Q), \tag{3.10}$$

where the set $\mathcal{M}$ is described as follows:

$$\mathcal{M} = \{Q \in \mathbb{R}^{n \times n} : Q = \sum_{(i,j) \in E} \frac{p_{ij}}{L_{ij}} Q_{ij},\ p_{ij} = p_{ji},\ p_{ij} = 0 \text{ if } (i,j) \notin E,\ \sum_{(i,j) \in E} p_{ij} = 1\}.$$

THEOREM 3.4. *Under the assumptions of Theorem 3.3 the optimal probabilities* $[p_{ij}^*]_{(i,j) \in E}$ *for achieving the best convergence rate in* (3.5) *is obtained by solving the SDP:*

$$[p_{ij}^*]_{(i,j) \in E} = \arg\max_{t,\ Q} \left\{ t : \quad Q + t\frac{aa^T}{a^T a} \succeq tI_n,\ \ Q \in \mathcal{M} \right\}. \tag{3.11}$$

Proof. First, we note that the following equivalence holds:

$$Q + t\frac{aa^T}{a^T a} \succeq tI_n \quad \text{if and only if} \quad t \leq \lambda_2(Q), \tag{3.12}$$

since the eigenvalues of the matrix $Q + \zeta aa^T$ are $\{\zeta a^T a, \lambda_2(Q), \cdots, \lambda_n(Q)\}$. Then, the optimization problem (3.10) can be written as $\max_{t,\ Q \in \mathcal{M},\ t \leq \lambda_2(Q)} t$, and combining with the LMI (3.12) we arrive at the SDP (3.11). Since this matrix Q depends linearly on p_{ij}, we can solve the optimization problem (3.11) as an SDP in the variables p_{ij} and obtain the optimal solution p_{ij}^* for all $(i,j) \in E$. ∎

3.3. Numerical experiments. We consider the following test problem (sometimes called Google problem) [7]: let $\bar{A} \in \mathbb{R}^{n \times n}$ be the incidence matrix of a graph G. Define $A = \bar{A}\,\mathrm{diag}(\bar{A}^T e)^{-1}$, where e is the vector with all entries equal to 1. Since $A^T e = e$, the goal is to determine a vector $x^* \geq 0$ such that:

$$Ax^* = x^* \quad \text{and} \quad e^T x^* = 1.$$

Clearly this problem can be written in optimization form: $\min_{x \in \mathbb{R}^n :\ e^T x = 1} f(x)$, where $f(x) = \|Ax - x\|^2$.

If we assume that the degree m_i of each node i in the graph is small compared to dimension of the problem n, then the computation of the partial derivatives of f is cheap (see also [7]). Indeed, if we define the matrix

$$W = [w_1 \cdots w_n] = A - I_n$$

and the residual

$$r(x) = Wx,$$

then

$$\nabla_i f(x) = w_i^T r(x).$$

Note that if $r(x)$ is already computed, then the computation of $\nabla_i f(x)$ requires $\mathcal{O}(m_i)$ operations. On the other hand, the update

$$x^+ = x + \alpha_i e_i + \alpha_j e_j$$

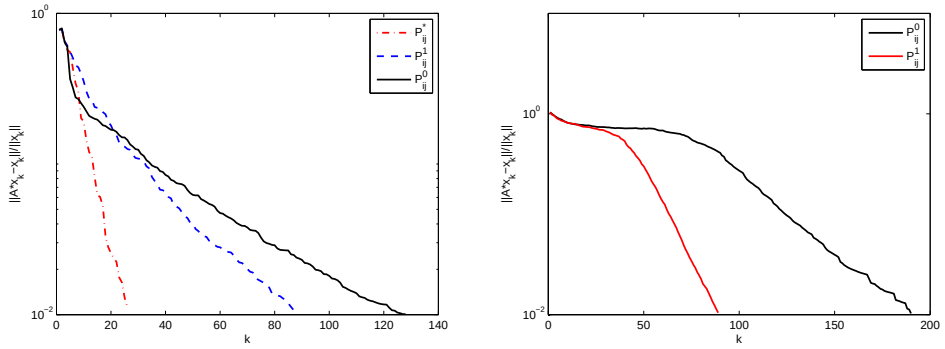

Fig. 3.1: *Randomly generated graph with average degree $m_i = 5$ (left) and $m_i = 20$ (right). Equivalent number of full iterations versus $\|Ax_k - x_k\|/\|x_k\|$. Left: $n = 30$ using probabilities p^0_{ij}, p^1_{ij} and p^*_{ij} . Right: $n = 10^5$ using probabilities p^0_{ij} and p^1_{ij}.*

implies the following change in the residual:

$$r(x^+) = r(x) + \alpha_i w_i + \alpha_j w_j.$$

In conclusion, the (i, j) iteration of method (RCD) needs $\mathcal{O}(m_i+m_j)$ operations that is much smaller than the computation of the whole gradient which requires $\mathcal{O}(\sum_{i=1}^n m_i)$ and is completely distributed, i.e. only two neighboring nodes $(i, j) \in E$ need to communicate at this iteration.

4. Conclusion. In this paper we developed a random coordinate descent algorithm for singly linear constrained convex optimization problems with expected rate of convergence $\mathcal{O}(\frac{1}{\lambda_2(Q)k})$, where $\lambda_2(Q)$ is the second smallest eigenvalue of a matrix Q that depends on the the choice of the probabilities and the number of blocks. However, the complexity per iteration of our method can be much simpler than of a method based on full gradient information. We also show how to design the probabilities such that the method converges as fast as possible. Finally, we presented preliminary computational results that demonstrate the numerical efficiency of our method.

REFERENCES

[1] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, 1999.
[2] C. Hsieh, K. Chang, C. Lin, S. Keerthi and S. Sundararajan, *A dual coordinate descent method for large-scale linear svm*, Proceedings of ICML, 408–415, 2008.
[3] Y. Li and S. Osher, *Coordinate descent optimization for l_1 minimization with application to compressed sensing: a greedy algorithm*, Inverse Problems and Imaging, 3, 487–503, 2009.
[4] A. A. Canutescu and R. L. Dunbrack, *Cyclic coordinate descent: a robotics algorithm for protein loop closure*, Protein Science, 12, 963–972, 2003.
[5] Z. Luo and P. Tseng, *A coordinate gradient descent method for nonsmooth separable minimization*, J. of Opt. Theory and Applications, 72 (1), 2002.
[6] I. Necoara, *A random coordinate descent algorithm for smooth convex optimization with coupled linear constraints*, submitted to IEEE Transactions on Automatic Control, 2012 (www.arxiv.org).
[7] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, Core Discussion Paper, 2010.

[8] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Boston, Kluwer, 2004.
[9] P. Richtarik and M. Takac, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, submitted to Mathematical Programming, 2011.
[10] L. Hurwicz, *The design of mechanisms for resource allocation*, American Economic Review, 63, 1–30, 1973.
[11] J. Kurose and R. Simha, *Microeconomic approach to optimal resource allocation in distributed computer systems*, IEEE Transactions on Computers, 38, 705-717, 1989.
[12] L. Xiao and S. Boyd, *Optimal Scaling of a gradient method for distributed resource allocation*, J. of Opt. Theory and Applications, 129, 2006.