Parallel Newton Methods for Numerical Analysis of Infeasible Linear Programming Problems with Block-Angular Structure of Constraints

Leonid D. Popov

[1] Krasovskii Institute of Mathematics and Mechanics UB RAS, Ekaterinburg, Russia
[2] Ural Federal University, Ekaterinburg, Russia
`popld@imm.uran.ru`

Abstract. For the linear programming (LP) problems, perhaps infeasible, with block-angular matrices of constraints, two parallel second order optimization methods are proposed. Being applied to initial LP problem they give its solution if this problem is solvable, and automatically deliver a solution of some relaxed LP problem, otherwise. The methods use penalty and barrier functions as well as Tikhonov regularization of Lagrangian of initial problem. The methods contain a parameter which tends to zero and minimize a residual of constraints. Parallel calculations of matrix operations follow MPI-type paradigm.

Keywords: linear programming, infeasibility, ill-posed problems, generalized solution, regularization, penalty function, parallel calculations

Introduction

Infeasible linear programs is widely studied class of the improper problems in mathematical programming theory (see [1,2,3,4,5,6] and others). They arise naturally in many applications of linear programming (LP) such as network flows analysis problems, problems of portfolio selection and scheduling problems which reflect complex production processes, e.g. in manufacture and power systems. Their inherent infeasibility may be caused by multiple factors such as errors in the input data, imbalance of its resource constraints and objectives, modelling bias as well as its structural distortions.

Basic approach addressing this issue is to introduce a more general form of solution for unsolvable problem which is in fact an optimum for some repaired or relaxed LP problem [1,2,7,8,9]. Following this approach in [10,11] two numerical methods were proposed which automatically adjust rigth-hand-side vector of initial LP problem if it is infeasible and then find its generalized solution using penalty and barrier functions as well as Tikhonov regularization of standard Lagrangian [12,13,14]. The methods contain a small parameter which tends to zero and minimize the residual of constraints.

In this paper we discuss how these methods may be fit to parallel calculations in MPI-type paradigm under the assumption that constraints matrix of initial LP problem has a block-angular structure.

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

1 The Problem State and Preliminaries

Consider the linear program

$$\min\{(c,\,x):\ Ax=b,\ x\geq 0\} \tag{1}$$

and its dual one

$$\max\{(b,\,y):\ A^{\top}y\leq c\}\ ; \tag{2}$$

where $A\in\mathbb{R}^{m_0\times n_0}$, $c\in\mathbb{R}^{n_0}$ and $b\in\mathbb{R}^{m_0}$ are given, $x\in\mathbb{R}^{n_0}$ and $y\in\mathbb{R}^{m_0}$ are vectors of primal and dual variables respectively, $\operatorname{rank}A=m_0\leq n_0$, $(\,\cdot\,,\,\cdot\,)$ denotes scalar product.

Assume that dual problem (2) is feasible, but it isn't known a priory whether primal problem (1) is feasible or not. If not then it may be transformed (or repaired) to solvable one merely by adjusting of its right-hand-side vector. Note that such problems are known as improper ones of the first kind [1,2].

If problem (1) is infeasible, we define its generalized solution as a solution $\hat{x}$ of the relaxed problem

$$\min\{(c,\,x):\ Ax=b+\varDelta\hat{b},\ x\geq 0\}\ (=:\hat{v})\ , \tag{3}$$

where $\varDelta\hat{b}$ =argmin$\{\|\varDelta b\|\colon Ax=b+\varDelta b,\ x\geq 0\}$, $\|\cdot\|$ is Euclidean norm. Obviously, the set of feasible solutions for (3) coincides with

$$M=\operatorname{Arg}\min_{x\geq 0}\|Ax-b\|\ .$$

If (1) is solvable, then $\varDelta\hat{b}=0$, and generalized solution introduced above coincides with an usual one.

As mentioned above, in [10,11] two numerical methods were proposed which being applied to LP problem (1) give its solution if such solution exists, or automatically deliver its generalized solution $\hat{x}$, if it is not so. Below we discuss how they may be fit to parallel calculations in MPI-type paradigm under the assumption that constraints matrix in (1) has a block-angular structure.

Two types of infeasible constraint angularity are investigated in this paper:
a) block-angular matrix with linking columns and
b) block-angular matrix with linking rows.
Both variants exploit parallel algebraic operations with matrices.

2 Block-Angular Matrices with Linking Columns

Consider the case when constraints matrix in infeasible problem (1) has the following structure

$$A=\begin{bmatrix} Q_1 & & & & & P_1\\ & Q_2 & & & & P_2\\ & & \ddots & & & \vdots\\ & & & Q_{n-1} & & P_{n-1}\\ & & & & Q_n & P_n \end{bmatrix}\ . \tag{4}$$

To take the advantage of this structure for parallelization of the calculation process, make use of the mixed barrier-penalty-type function for problem (1)

$$\varPhi(\epsilon; x) = (c, x) + \frac{1}{2\epsilon}\,\|Ax - b\|^2 - \epsilon \sum_{i=1}^{n_0} \ln x^i, \quad \epsilon > 0 \ . \tag{5}$$

Consider the unconstrained smooth optimization problem: find $\hat{x}_\epsilon > 0$ such that

$$\varPhi(\epsilon; \hat{x}_\epsilon) = \min_{x>0} \varPhi(\epsilon; x) \ . \tag{6}$$

As shown in [10], if optimal set of (3) is bounded, then an unique minimizer $\hat{x}_\epsilon$ exists for every $\epsilon > 0$, even in improper case. This minimizer satisfies to nonlinear vector equation

$$\nabla_x \varPhi(\epsilon; x) = c + \epsilon^{-1} A^T (Ax - b) - \epsilon X^{-1} e = 0 \ , \tag{7}$$

where X is diagonal matrix with coordinates x^i on its diagonal, i.e. $X = \mathrm{diag}(x)$, $e = [1, \ldots, 1]^T$.

Theorem 1 **(**[10]**).** *Suppose that the feasible set of the dual problem* (2) *has an interior point. Then*

$$(c, \hat{x}_\epsilon) \to \hat{v}, \quad b - A\hat{x}_\epsilon = \varDelta\hat{b}_\epsilon \to \varDelta\hat{b}$$

as $\epsilon \to +0$.

Corollary 1. *Let the optimal vector $\hat{x}$ of the relaxed problem* (3) *be unique. Then $\hat{x}_\epsilon \to \hat{x}$ as $\epsilon \to +0$.*

To solve equation (7), one can apply Newton method as follows

$$x(t+1) = x(t) - \alpha_t w(t), \quad w(t) = H^{-1}(\epsilon; x(t)) \nabla \varPhi(\epsilon; x(t)), \quad t = 0, 1, \ldots \ .$$

Here $x(0)$ is an appropriate start point, α_t is a step parameter (usually $\alpha_t \equiv 1$), $H(\epsilon; \cdot)$ is a Hessian of the function $\varPhi(\epsilon; \cdot)$, $\nabla_x \varPhi(\epsilon; \cdot)$ is its gradient.

Without going deep into all issues of implementation of this algorithm (e.g. see [15] for details), we only consider how to calculate vector $w(t)$ in parallel. In our case

$$H(\epsilon; x) = \epsilon^{-1} A^T A + \epsilon X^{-2} \ .$$

Hence, to find $w(t)$ one has to solve a sparse linear system

$$H(\epsilon; x(t)) w = \nabla \varPhi(\epsilon; x(t)) \tag{8}$$

with

$$H(\epsilon; x) = \begin{pmatrix} H_1 & & & & M_1^T \\ & H_2 & & & M_2^T \\ & & \ddots & & \vdots \\ & & & H_n & M_n^T \\ M_1 & M_2 & \ldots & M_n & M_{n+1} \end{pmatrix} ,$$

where

$$H_i = \epsilon^{-1} Q_i^T Q_i + \epsilon X_i^{-2} \ (i = 1, \dots, n), \quad M_i = \epsilon^{-1} P_i^T Q_i \ (i = 1, \dots, n) \ ,$$

$$M_{m+1} = \epsilon^{-1} \sum_{i=1}^{n} P_i^T P_i + \epsilon X_{n+1}^{-2} \ .$$

Diagonal matrices $X_i = \operatorname{diag}(x_i)$ above correspond to the partition of a primal vector x onto sub-vectors x_i according to matrix partition (4).

Sparse structure of the Hessian gives us the opportunity to solve linear system (8) in parallel mode (see section 4 for details).

3 Block-Angular Matrices with Linking Rows

Now consider the case when

$$A = \begin{bmatrix} Q_1 & & & & \\ & Q_2 & & & \\ & & \ddots & & \\ & & & Q_{n-1} & \\ & & & & Q_n \\ P_1 & P_2 & \dots & P_{n-1} & P_n \end{bmatrix} \ . \tag{9}$$

To exploit this structure for parallelization of the calculation process, make use of the regularized barrier-type function for problem (2)

$$\Psi(\epsilon; x) = (b, y) - \frac{\epsilon}{2} \|y\|^2 + \epsilon \sum_{i=1}^{n_0} \ln (c^i - (A_i, y)), \quad \epsilon > 0 \ , \tag{10}$$

where A_i denotes the i-th column of the matrix A. Function (10) is finite and strongly concave on $\Omega = \{y \colon A^T y < c\}$. Therefore, for every $\epsilon > 0$ there exists an unique vector $\hat{y}_\epsilon$ such that

$$\Psi(\epsilon; \hat{y}_\epsilon) = \max_{y \in \Omega} \Psi(\epsilon; y) \ . \tag{11}$$

Note that $\hat{y}_\epsilon$ satisfies to nonlinear vector equation

$$\nabla_y \Psi(\epsilon; \hat{y}_\epsilon) = b - \epsilon \hat{y}_\epsilon + \epsilon A D(y)^{-1} e = 0 \ , \tag{12}$$

where diagonal matrix $D(y) = \operatorname{diag}(c - A^T y)$ has elements $d^i = c^i - (A_i, y)$ on its diagonal, $e = [1, \dots, 1]^T$.

Theorem 2 ([11]). *Let $\hat{y}_\epsilon$ be the maximizer of $\Psi(\epsilon; \cdot)$ on y. Then the vector $\hat{x}_\epsilon$ with coordinates*

$$\hat{x}_\epsilon^i = \epsilon (c^i - (A_i, \hat{y}_\epsilon))^{-1}, \quad i = 1, \dots, n_0 \ ,$$

is the minimizer of $\Phi(\epsilon; \cdot)$ on $x > 0$.

Corollary 2. *Suppose that the feasible set of the dual problem* (2) *has an interior point. Then* $\epsilon\hat{y}_\epsilon \to \Delta\hat{b}$ *as* $\epsilon \to +0$.

To solve equation (12) one can apply Newton method again. Now its iterations are as follows

$$y(t+1) = y(t) + \alpha_t w(t), \quad w(t) = H^{-1}(\epsilon; y(t))\nabla_y \Psi(\epsilon; y(t)), \quad t = 0, 1, \ldots \ .$$

Here $y(0)$ is an appropriate start point, α_t is a step parameter (usually $\alpha_t \equiv 1$), $H(\epsilon; \cdot)$ is a Hessian of the function $\Psi(\epsilon; \cdot)$, $\nabla_y \Psi(\epsilon; \cdot)$ is its gradient.

In the case under consideration

$$H(\epsilon; y) = \epsilon I + \epsilon A D(y)^{-2} A^T \ \ ,$$

where I is the identity matrix of appropriate order. Therefore, to find $w(t)$ one has to solve a sparse linear system

$$H(\epsilon; y(t)) w = \nabla_y \Psi(\epsilon; y(t)) \tag{13}$$

with

$$H(\epsilon; x) = \begin{pmatrix} H_1 & & & & M_1^T \\ & H_2 & & & M_2^T \\ & & \ddots & & \vdots \\ & & & H_n & M_n^T \\ M_1 & M_2 & \ldots & M_n & M_{n+1} \end{pmatrix} \ ,$$

where

$$H_i = \epsilon I_i + \epsilon Q_i D_i(y)^{-2} Q_i^T, \ \ M_i = \epsilon P_i D_i(y)^{-2} Q_i^T, \quad (i = 1, \ldots, n) \ \ ,$$

$$M_{n+1} = \epsilon I_{n+1} + \epsilon \sum_{i=1}^{n} P_i D_i(y)^{-2} P_i^T \ \ .$$

Here, diagonal matrices $D_i(y) = \mathrm{diag}(c_i - Q_i^T y_i - P_i^T y_{n+1})$ correspond to the partition of a dual vector y onto sub-vectors y_i according to matrix partition (9).

Again sparse structure of the Hessian gives us the opportunity to solve linear system (13) in parallel mode.

4 Parallel Scheme of Implementation

It is easy to see that in the both cases above the structure of the Hessians is just the same. Let us discuss how to solve a sparse linear system of the type

$$\begin{pmatrix} H_1 & & & & M_1^T \\ & H_2 & & & M_2^T \\ & & \ddots & & \vdots \\ & & & H_n & M_n^T \\ M_1 & M_2 & \ldots & M_n & M_{n+1} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \\ f_{n+1} \end{pmatrix} \tag{14}$$

on massively parallel processor (MPP) or cluster of workstations (COW) with MPI-instructions.

For simplicity assume that a number of processors in our MIMD-computer is equal to $n+1$. Let processors with indexes from 1 to n be called *the slaves* and processor with index $n+1$ be called *the master*. Suppose that input data of LP problem (1) is distributed between the slaves in such a manner that the slave with index i stores the matrix pair $(Q_i; P_i)$ in its local memory. Therefore, each slave with index i can calculate its own part of Hessian $(H_i; M_i)$ and some auxiliary matrices with index i concurrently. The master controls the calculation process and, besides, calculates H_{n+1} and other matrices with additional index $n+1$. All processors exchange with the messages using MPI-instructions like *send*, *gather* and *broadcast* to coordinate their works.

Parallel resolution of system (14) may be as follows. At first, all the slaves calculate its own Cholesky decomposition of $H_i = L_i L_i^T$ concurrently, then form the auxiliary matrices $S_i = L_i^{-1} M_i^T$, $R_i = S_i^T S_i$, and send them to the master. The master gathers this matrices into a global sum $\bar{H}_{n+1} = H_{n+1} - \sum_{i=1}^n R_i$ and then calculates its own Cholesky decomposition of $\bar{H}_{n+1} = L_{n+1} L_{n+1}^T$. As a result the global Cholesky decomposition is calculated

$$\begin{pmatrix} H_1 & & & & M_1^T \\ & H_2 & & & M_2^T \\ & & \ddots & & \vdots \\ & & & H_n & M_n^T \\ M_1 & M_2 & \dots & M_n & M_{n+1} \end{pmatrix} = \begin{pmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_n & \\ S_1^T & S_2^T & \dots & S_n^T & L_{n+1} \end{pmatrix} \begin{pmatrix} L_1^T & & & & S_1 \\ & L_2^T & & & S_2 \\ & & \ddots & & \vdots \\ & & & L_n^T & S_n \\ & & & & L_{n+1}^T \end{pmatrix} .$$

Further, all the slaves solve their own angular systems $L_i z_i = f_i$ concurrently, then form the products $h_i = S_i^T z_i$ and send them to the master. The master gathers this vectors, forms the global sum $\bar{f}_{n+1} = f_{n+1} - \sum_{i=1}^n h_i$ and solves its own angular system $L_{n+1} z_{n+1} = \bar{f}_{n+1}$. Thereby, a solution of the following system is obtained

$$\begin{pmatrix} L_1 & & & & \\ & L_2 & & & \\ & & \ddots & & \\ & & & L_n & \\ S_1^T & S_2^T & \dots & S_n^T & L_{n+1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \\ f_{n+1} \end{pmatrix} .$$

Finally, the master solves the angular system $L_{n+1}^T w_{n+1} = z_{n+1}$ and broadcasts w_{n+1} to all the slaves. The slaves solve their angular systems $L_i^T w_i = z_i - S_i w_{n+1}$ concurrently. Therefore, the following system is solved

$$\begin{pmatrix} L_1^T & & & & S_1 \\ & L_2^T & & & S_2 \\ & & \ddots & & \vdots \\ & & & L_n^T & S_n \\ & & & & L_{n+1}^T \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1} \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \\ z_{n+1} \end{pmatrix} .$$

It means that all components of $w = [w_1, \dots, w_n, w_{n+1}]$ are found.

Note that similar parallel scheme of calculations was applied by author for sparse linear system in [16], where one can find the estimates of possible speedup

$$S_{n+1} \approx \frac{n}{1+Cm^{-1}} \to n \quad (m \to \infty) \ , \tag{15}$$

as well as an example of implementation scheme for Parallel Matlab and the encouraging results of numerical experiments with large LP problems. In (15) $C > 0$ is some technical constant depending upon concrete computer equipment, m characterizes the average dimension of the blocks of the matrix A in problem (1). If m is comparatively large then the speedup tends to n.

5 Conclusion

For a linear program with block-angular matrix of constraints, two parallel second order optimization methods are proposed. The methods give usual solution if initial problem is solvable, and automatically deliver a solution of some relaxed LP problem, otherwise. The methods use penalty and barrier functions as well as Tikhonov regularization of Lagrangian of initial LP problem. The methods contain a small parameter which tends to zero and minimize the residual of constraints. Parallel calculations of matrix operations follow MPI-type paradigm.

Acknowledgements. This work was supported by Russian Foundation for Basic Research (project no. 16-07-00266).

References

1. Eremin, I. I., Mazurov, Vl. D., and Astaf'ev, N. N.: Improper Problems of Linear and Convex Programming. Nauka, Moscow. In Russian (1983)
2. Eremin, I. I. Theory of Linear Optimization. Inverse and Ill-Posed Problems Series. VSP. Utrecht, Boston, Koln, Tokyo (2002)
3. Chinneck, J.W.: Feasibility and viability. in: T. Gal, H.J. Greenberg (Eds.), Advances in Sensitivy Analysis and Parametric Programming, Kluwer Academic Publishers, Boston (1997)
4. Ben-Tal, A.; Nemirovski, A.: Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data. Math. Progr. 88: 3, 411–424 (2000)
5. Amaral, P.; Barahona, P.: Connections between the total least squares and the correction of an infeasible system of linear inequalities. Linear Algebra and its Appl. 395, 191–210 (2005)
6. Leon, T; Liern, V.: A Fuzzy Method to Repair Infeasibility in Linear Constrained Problems. Fuzzy Set and Systems. 122, 237–243 (2001)
7. Dax, A.: The smallest correction of an inconsistent system of linear inequalities. Optimization and Engineering. 2, 349–359 (2001)
8. Eremin, I. I.; Popov, L.D.: Numerical analysis of improper linear programs using the DELTA-PLAN-ES interactive system. Optimization Methods and Software. Vol.2. 69–78 (1993)

9. Skarin, V. D.: Regularized Lagrange function and correction methods for improper convex programming problems. Proc. Steklov Inst. Math. Mathematical Programming. Regularization and Approximation, suppl. 1, S116–S144 (2002)
10. Popov, L. D.: Use of barrier functions for optimal correction of improper problems of linear programming of the 1st kind. Automation and Remote Control. 73:3, 417–424 (2012)
11. Popov, L. D.: Dual approach to the application of barrier functions for the optimal correction of improper linear programming problems of the first kind. Proceedings of the Steklov Institute of Mathematics. 288:1, 173–179 (2015)
12. Fiacco, A.V.; McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. John Wiley and Sons, Inc: New York-London-Sidney (1968)
13. Tikhonov, A.N.; Vasil'ev, F.P.: Methods for solving ill-posed extremal problems, in: Mathematics Models and Numerical Methods. Banach center publ., Warszava. 3, 291–348 (1978)
14. Rockafellar, R.T.: Convex Analysis. Princeton University Press: Princeton, N.J. (1970)
15. Gill, Philip E.: Practical optimization / Philip E. Gill, Walter Murray, Margaret H. Wright. London ; New York : Academic Press (1981)
16. Popov, L. D.: Experience in organizing hybrid parallel calculations in the Evtushenko–Golikov method for problems with block-angular structure. Automation and Remote Control. 75:4, 622–631 (2014)