Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 10, n. 1, 2023.

Trabalho apresentado no XLII CNMAC, Universidade Federal de Mato Grosso do Sul - Bonito - MS, 2023

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Handling dense columns in Interior-Point Methods

Catalina J. Villalba[1], Aurelio R.L. Oliveira[2]
UNICAMP, Campinas, SP

Abstract. The Interior-Point methods are a type of method used to solve linear programming problems that require solving linear systems. In situations where the constraint matrix has dense columns, it is essential to find an efficient way to solve computationally these systems in order to avoid memory issues or increase the number of operations. This project proposes a preconditioner to handle this issue, and it provides both theoretical predictions and computational tests to demonstrate its effectiveness.

Key words. Linear programming, Interior-point methods, Preconditioner.

1 Introduction

Interior-point methods (IPM) applied to solve linear programming problems give raise to linear systems to be solved at each iteration [5] and there are cases in which the constraint matrix contains at least one dense column, that is, it has many non-zero elements. Not considering dense and sparse columns separately may deliver linear systems with almost full matrices, which implies a large number of floating point operations to be performed or even memory difficulties to deal with large-scale problems.

Some studies have focused on the analysis of linear programming problems whose matrix has these characteristics. These proposals include the modification of the constraint matrix A, splitting each dense column into several sparse columns [6], the creation of an equivalent augmented system [8], the modification of the Schur's Complement [1] and a modification of the Cholesky factorization of the sparse matrix [4].

This project proposes the use of a preconditioner applied to an equivalent system for this type of problem. We prove theoretically that the final system is uniformly bounded when the IPM is converging and this result is computationally verified when compared to other approaches.

2 Linear programming

The objective of a linear programming problem (LP) is to optimize a linear function subject to linear constraints. The **standard form** is given by

$$\begin{aligned} \min \ & c^T x \\ \text{s.t. } & Ax = b \\ & x \geq 0, \end{aligned} \tag{1}$$

[1]catajv@ime.unicamp.br
[2]aurelio@unicamp.br

where $A \in \mathbb{R}^{m \times n}$ has full row-rank, i.e., $rank(A) = m \leq n$; $c, x \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. This LP problem is known as primal (PP). Each (PP) has associated a dual problem (DP):

$$\begin{aligned} &\max\ b^T y \\ &\text{s.t. } A^T y + z = c \\ &\quad\ y \text{ free}, z \geq 0. \end{aligned} \tag{2}$$

The optimality conditions for (PP) and (DP) are given by the Karush-Kuhn-Tucker (KKT) theorem from [5].

Theorem 2.1 (KKT optimality conditions)**.** *x and (y, z) are optimal solutions to the primal and dual problems, if and only if*

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \Big\} \quad \textit{Primal feasible}$$

$$\begin{aligned} A^T y + z &= c \\ z &\geq 0 \end{aligned} \Big\} \quad \textit{Dual feasible} \tag{3}$$

$$XSe = 0 \ \Big\} \quad \textit{Complementary}$$

where X and S are diagonal matrices, whose entries are the elements of vectors x and s respectively; and e is a vector of ones with the right dimension.

Since computing the exact solution may be costly, we can relax the complementary condition of the KKT theorem by introducing a scalar parameter $\mu \in (0, 1)$.

$$XZe = \mu e. \tag{4}$$

The complementarity is a nonlinear condition, therefore, in order to compute the optimal solution is possible to employ Newton's method. To obtain the search directions $(\Delta x, \Delta y, \Delta z)$, it is necessary to solve the following linear system at each k iteration of IPM:

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta z^k \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - z \\ \mu e - XZe \end{bmatrix} := \begin{bmatrix} r_b^k \\ r_c^k \\ r_{xz}^k \end{bmatrix}. \tag{5}$$

By eliminating variables, we obtain the *normal equations*:

$$A\Theta^k A^T \Delta y^k = r_b^k - A\Theta^k \left[(X^k)^{-1} r_{xz}^k - r_c^k) \right], \tag{6}$$

where

$$\Theta^k = (Z^k)^{-1} X^k. \tag{7}$$

Since A is a full-rank matrix, the normal equation has a symmetric positive definite matrix. Therefore, direct methods like Cholesky Factorization or iterative approaches like Conjugate Gradient (or Preconditioned Conjugate Gradient) can be used to solve the linear system. From now we will eliminate the index k related to the iteration of IPM.

So far we discussed the process to solve a linear programming problem with Interior-Point methods. Now, analyzing the structure of the constraint matrix, a column is considered dense if it has more than a certain number of non-zero elements. Let P be a column permutation matrix such that $AP = [S, D]$, where $D : m \times k$ contains the k dense columns and $S : m \times (n - k)$ the sparse columns. Also, for scale matrix we have $P\Theta P^T = diag(\Theta_D, \Theta_S)$. Supposing $P = I_n$, the normal equations split into two as following

$$A\Theta A^T = S\Theta_S S^T + D\Theta_D D^T. \tag{8}$$

In order to improve the numerical stability or to get an easier system to be solved it is feasible to apply preconditioners.

DOI: 10.5540/03.2023.010.01.0060

3 Preconditioner

The aim of using preconditioners is to improve the numerical stability of the problems to be solved and to reduce the number of operations to be performed in each iteration of IPM. Thus, starting from the splitting of A into dense and sparse columns, we create an augmented linear system with matrix K, which is equivalent to the one delivered for each iteration of IPM. K is an indefinite matrix with larger dimension and more sparse than the original system delivered of IPM. Let M be a nonsingular matrix, the preconditioned system is $M^{-1}KM^{-T}\tilde{w} = \tilde{r}$, where $\tilde{w} = M^T w$ and $\tilde{r} = M^{-1}r$. The adequate definition of M makes the preconditioned matrix more stable and a final system easier to solve with respect to the original one.

Let $F : m \times d$, $d \leq k$ be such that $LL^T = S\Theta_S S^T + FF^T$, where L is a lower triangular matrix. We propose a preconditioner applied to the matrix

$$K = \begin{bmatrix} S\Theta_S S^T + FF^T & D\Theta_D^{1/2} & F \\ \Theta_D^{1/2} D^T & -I_k & 0 \\ F^T & 0 & I_d \end{bmatrix}. \tag{9}$$

The linear system

$$K\omega = K \begin{bmatrix} \Delta y \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \hat{r} \\ 0 \\ 0 \end{bmatrix} = \tilde{r} \tag{10}$$

is equivalent to (6); where $q_1 : k \times 1$ and $q_2 : d \times 1$. Let

$$M^{-1} = \begin{bmatrix} L^{-1} & L^{-1}D\Theta_D^{1/2} & -L^{-1}F \\ 0 & \Theta_D^{-1/2} & 0 \\ 0 & 0 & I_k \end{bmatrix}, \tag{11}$$

where I_k is the identity matrix $k \times k$ and L is the Cholesky factor of the sparse matrix $S\Theta_S S^T + FF^T$. Define $G := L^{-1}D\Theta_D^{1/2}$ and $J := -L^{-1}F$. Then, the preconditioned matrix is

$$M^{-1}KM^{-T} = \begin{bmatrix} I_m + GG^T - JJ^T & 0 & 0 \\ 0 & -\Theta_D^{-1} & 0 \\ 0 & 0 & I_d \end{bmatrix}. \tag{12}$$

Therefore, to recover the search direction Δy it is necessary to solve a linear system involving the matrix $W = I_m + GG^T - JJ^T$. Since the matrices G and J depend on Θ, we would like to analyze the behavior of the matrix W when IPM is converging to the optimal solution.

4 Numerical stability

Definition 4.1. *A matrix $B : m \times n$ the that depends on γ is **uniformly bounded** if there is a constant c independent of γ such that $|b_{ij}(\gamma)| \leq c$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$. We say $B = \mathcal{O}(1)$. This means that the matrix B remains relatively constant as γ changes.*

In our particular case, we proceed to analyze that matrix W is uniformly bounded when γ is converging to zero.

Definition 4.2. *A matrix $B : m \times n$ that depends on γ is $\Theta(\gamma^2)$ if there are positive constants c_1 and c_2 such that*

$$c_1\gamma^2 \leq |b_{ij}(\gamma)| \leq c_2\gamma^2, \qquad i = 1 \ldots m, j = 1 \ldots n. \tag{13}$$

Proposition 4.1. *$W = I_m + GG^T - JJ^T$ is positive definite.*

Proof.

$$\begin{aligned} W &= I + GG^T - JJ^T \\ &= I + L^{-1}D\Theta_D D^T L^{-T} - L^{-1}FF^T L^{-T} \\ &= L^{-1}(LL^T + D\Theta_D D^T - FF^T)L^{-T} \\ &= L^{-1}(A\Theta A^T)L^{-T}. \end{aligned} \tag{14}$$

Since A is full rank, Θ is a positive diagonal matrix and L is the Cholesky factor of the sparse part, therefore W is a positive definite matrix. □

The next two theorems from Goldfarb, D. and Scheinberg, K. in [4] are important to analyze the numerical stability of matrix W.

Theorem 4.1. *Let $A : m \times n$ and Θ be a positive diagonal matrix, whose elements depend on γ and $L\Sigma L^T$ be the Cholesky factorization of $A\Theta A^T$, where L is a lower triangular matrix with ones on the main diagonal and Σ is a positive diagonal matrix. Then, the entries of L are uniformly bounded when γ is approaching zero and the diagonal entries of the matrix Σ are either $\Theta(\gamma)$ or $\Theta(\gamma^{-1})$.*

Theorem 4.2. *If L is a lower triangular matrix with ones on the diagonal, and the subdiagonal entries that depend on a parameter γ are such that they are uniformly bounded, then the entries of inverse L^{-1} are also uniformly bounded.*

Based on the previous theorems, we can use the facts that L is the Cholesky factor of the sparse matrix $S\Theta_S S^T$, also that $D\Theta_D D^T$ is semidefinite positive, the structure and the definition of F, and the asymptotic properties to conclude the two following propositions related to the matrices $GG^T = L^{-1}D\Theta_D D^T L^{-T}$ and $JJ^T = -L^{-1}FF^T L^{-T}$:

Proposition 4.2. *$GG^T = \mathcal{O}(1)$ and $JJ^T = \mathcal{O}(1)$.*

Proposition 4.3. *W is uniformly bounded.*

Proof. For each $i, j \in \{1, \ldots, m\}$, we have

$$(W)_{ij} = \begin{cases} (GG^T)_{ij} - (JJ^T)_{ij} & \text{if } i \neq j, \\ 1 + (GG^T)_{ij} - (JJ^T)_{ij} & \text{if } i = j. \end{cases} \tag{15}$$

Since GG^T and FF^T are uniformly bounded when IPM is close to an optimal solution, by definition it follows that $W = \mathcal{O}(1)$. □

5 Computational tests

The numerical experiments were performed with the preconditioner defined in Section 3 and a modification of the PCx [3] algorithm. The process of the Interior-Point method is developed in C and the Cholesky factorization associated with the sparse matrix $S\Theta_S S^T + FF^T$ was developed in Fortran, based on the proposal for factoring sparse matrices of [7].

The computational tests were performed in the Linux environment, on an Intel Core i7-3770K, processor 3.50GHz with 32 GB RAM. For the present project, the LP problems of the article [2] were tested, for the particular case of Minimum-distance controlled perturbation methods with L-infinity norm (LP library in `http://www-eio.upc.es/~jcastro/`).

Our proposal, which we denote as PCx_mod, was compared with the original PCx process, (PCx_orig). Also, we compared with the Modified Schur-Complement of Andersen (PCx_Andersen) in [1]. For these 3 cases, we use the same density criteria. Likewise, we also analyze the case of not splitting dense and sparse columns (PCx_full). All of these were developed with a modification of PCx.

Table 1 contains the dimensions of the tested problems. These dimensions are after having performed preprocessing process. Nnz A refers to the number of nonzeros entries of matrix A, and ndense is the number of dense columns.

The running time for each case is summarized in Table 2. It does not include the time for preprocessing. We can observe that some problems produce results that are essentially the same for PCx_mod, PCx_orig, and PCx_Andersen. Our suggestion had better success in some cases. The PCx_full approach obtained extremely high times, as anticipated. Those that obtained "UNKNOWN" result is because the convergence rate is very slow or the gamma value increases from one iteration to another, and "-" refers that a problem took more than 5 hours to be solved.

And finally, Table 3 exposes the number of iterations required to find an optimal solution of IPM. Similar results are observed for some examples. This result is complementary to the running time analysis.

To solve the linear system associated with the matrix W, we applied the Conjugated Gradient (CG) iterative method and for each iteration of IPM it was required one iteration of CG.

Table 1: Tested problems.

LP problem	m	n	Nnz A	ndense
Linf_bts4	55145	96465	31150	1
Linf_five20c	47128	74279	23619	1
Linf_five20b	48143	76873	24506	1
Linf_jjtabeltest3	2563	4240	1686	2
Linf_nine5d	9782	15897	5024	1
Linf_nine12	15770	26795	8298	1
Linf_ninenew	9139	15342	4709	1
Linf_table3	3477	6453	1875	1

Table 2: Running time of each approaching method

	Time (s)			
LP problem	PCx_mod	PCx_orig	PCx_Andersen	PCx_full
Linf_bts4	**23.01**	25.86	24.26	110446.02
Linf_five20c	**1064.38**	1064.44	1065.45	51763.45
Linf_five20b	**189.13**	196.47	189.95	-
Linf_jjtabeltest3	0.06	**0.05**	INFEASIBLE	7.33
Linf_nine5d	**1.76**	1.77	1.77	334.5
Linf_nine12	**17.44**	17.69	17.91	1769.12
Linf_ninenew	**13.58**	13.91	14.24	406.35
Linf_table3	**1.11**	INFEASIBLE	UNKNOWN	34.91

DOI: 10.5540/03.2023.010.01.0060

Table 3: Iterations for each approaching method

	Number of iterations			
LP problem	PCx_mod	PCx_orig	PCx_Andersen	PCx_full
Linf_bts4	39	40	**34**	36
Linf_five20c	20	20	20	20
Linf_five20b	**20**	**20**	21	-
Linf_jjtabeltest3	28	28	INFEASIBLE	**24**
Linf_nine5d	16	16	16	**15**
Linf_nine12	14	14	14	15
Linf_ninenew	**13**	14	14	14
Linf_table3	27	INFEASIBLE	UNKNOWN	**24**

The omission of the dense columns leads to a noticeably longer running time compared to any method that includes them, as confirmed by the computational results. Our proposed preconditioner provides an advantage in terms of the rapid convergence of the Conjugate Gradient method when solving the linear system. This enables it to compete favorably with other approaches in terms of running time. Moreover, our method demonstrated the capability to solve a greater number of LP problems that were either infeasible or unknown to other approaches. This is evident from the results obtained for the *Linf_table3* problem, where our method successfully achieved the optimal solution.

6 Conclusions

This project presents a new approach for handling LP problems that involve dense columns, by utilizing a preconditioner on a modified system. Theoretical evidence supports the assertion that the resulting linear system remains uniformly bounded. Moreover, computational tests indicate that this method has the potential to enhance running time or reduce the number of iterations required.

To evaluate the effectiveness of the proposed preconditioner, it was competitive computationally against other existing approaches. The results of computational tests revealed comparable outcomes in terms of the number of iterations and running time. However, our proposal surpassed these approaches by successfully solving a greater number of LP problems, establishing its reliability. These findings emphasize the competitiveness of the proposed method as an effective approach for LP problems with dense columns.

Acknowledges

This study was partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico -Brazil (CNPq) - Grants 141611/2020-0 and 313258/2021-0.

References

[1] K. D. Andersen. "A modified Schur-complement method for handling dense columns in interior-point methods for linear programming". In: **ACM Transactions on Mathematical Software (TOMS)** 22.3 (1996), pp. 348–356. DOI: 10.1145/232826.232937.

[2] J. Castro. "Minimum-distance controlled perturbation methods for large-scale tabular data protection". In: **European Journal of Operational Research** 171.1 (2006), pp. 39–52. DOI: 10.1016/j.ejor.2004.08.034.

[3] J. Czyzyk et al. "PCx: an interior-point code for linear programming". In: **Optimization Methods and Software** 11.1-4 (1999), pp. 397–430. DOI: 10.1080/10556789908805757.

[4] D. Goldfarb and K. Scheinberg. "A product-form Cholesky factorization method for handling dense columns in interior point methods for linear programming". In: **Mathematical Programming** 99.1 (2004), pp. 1–34. DOI: 10.1007/s10107-003-0377-7.

[5] J. Gondzio. "Interior point methods 25 years later". In: **European Journal of Operational Research** 218.3 (2012), pp. 587–601. DOI: 10.1016/j.ejor.2011.09.017.

[6] J. Gondzio. "Splitting dense columns of constraint matrix in interior point methods for large scale linear programming". In: **Optimization** 24.3-4 (1992), pp. 285–297. DOI: 10.1080/02331939208843796.

[7] E. G Ng and B. W. Peyton. "Block sparse Cholesky algorithms on advanced uniprocessor computers". In: **SIAM Journal on Scientific Computing** 14.5 (1993), pp. 1034–1056. DOI: 10.1137/0914063.

[8] R. J. Vanderbei. "Splitting dense columns in sparse linear systems". In: **Linear Algebra and its Applications** 152 (1991), pp. 107–117. DOI: 10.1016/0024-3795(91)90269-3.

