International Journal of Pure and Applied Mathematics
Volume 60 No. 3 2010, 239-244

AN EFFICIENT SIMPLEX LU FACTORIZATION UPDATE

Daniela R. Cantane[1] §, Aurelio R.L. Oliveira[2]
[1,2]School of Electric Engineering and Computation (FEEC)
State University of Campinas (UNICAMP)
400, Albert Einsten Av., Campinas, 13083-852, SP, BRAZIL
[1]e-mail: dcantane@densis.fee.unicamp.br

Abstract: In this work we develop simplex basis LU update factorization techniques, using a static reordering of the matrix columns. In the static reordering, the matrix columns are rearranged in accordance with the increasing number of nonzero entries, leading to sparse factorization of the basis without computational effort to reorder the columns. Therefore the matrix reordering is static and the columns of the basis follow this ordering. A simulation of the simplex iterations is carried through according to the base sequence obtained from *MINOS*. The factorization and LU update are performed considering sparsity. The objective of this work is to compare the developed reordering approach with the results from *MINOS*. Preliminary computational results in *Matlab* for problems from the Netlib collection show that this is a very promising idea, since there is no need to refactorize the matrix in the tested problems.

AMS Subject Classification: 49N05
Key Words: linear optimization, sparse matrix, factorization update, simplex

1. Introduction

The efficient solution of large-scale linear systems is very important for solving linear optimization problems. These systems can be approached through generic methods as, for example, LU factorization of the basis and its update, or through the problem specific structure exploitation, as in the network flow problem (Kennington and Helgason [4]).

In Bressan and Oliveira [2] the column reordering in the combined cutting

Received: February 12, 2007

§Correspondence author

stock and lot sizing problems (Nonas and Thorstenson [7]) is done in such a way that the resulting matrix is block diagonal. Therefore, it is easy to factorize the base, resulting in little fill-in. Through the static reordering of columns, it is possible to obtain a sparse factorization of the basis, without any overhead to determine the order of the columns, since the base columns follow the given ordering.

In the LU update, the floating point operations originated by the column that leaves the base are undone, in the reverse order of the factorization, always considering sparsity. Finally, it is necessary to compute the factorized columns after the entering column, also considering the sparse pattern.

For the combined problem, results obtained in Bressan and Oliveira [2] have shown that this approach is fast and robust, introducing insignificant accumulated rounding errors in worst case situations, after thousands of iterations.

2. LU Factorization Update Methods

The LU factorization update technique with partial pivot was proposed by Bartels [1]. Two variants of this algorithm proposed in Reid [8] aim to balance the sparsity and numeric stability in the factorization. The latter variant is an improvement over the former.

In Maros [6], the LU factorization update proposed by Markowitz [5] and its application in the simplex update basis method is describe in detail. Moreover, it presents the ideas of an efficient implementation proposed by Suhl and Suhl [9], that is a good combination of the symbolic and numerical phase of the pivoting and presents a compromise between sparsity and numerical stability. Other update techniques are described in Duff et al [3].

3. Implementation Issues

The objective of the implementation developed here is to simulate simplex iterations using the static reordering and the LU update adopted in Bressan and Oliveira [2] for general linear optimization problems.

The base sequence obtained from *MINOS* is used in the simulation. A procedure for finding the initial basis, "Crash" base described in Maros [6] and the leaving and entering columns of the basis from *MINOS* output was implemented. Matrix columns are reordered according to the number of nonzero

entries, in increasing order, leading to sparse factorizations without computational effort to obtain the order of columns, since the reordering of the matrix is static and basis columns follow this ordering.

Two column update approaches are used in the implementation and the number of nonzero entries for each one is compared. When the entering column e in the basis follows the static ordering the approach is called U_1. In the second approach, called U_2, the entering column e entries the basis in the position of the leaving column l. The notation U_{minos} is the used for *MINOS* results.

In both approaches, a LU factorization of the basis considering its sparsity is done. Only the columns factorization actually modified by the change of the basis are carried through. Notice that there is no changes in column k located after the entering and/or leaving column if $LU(k, e) = 0$ or $LU(k, l) = 0$ due to sparse structure of columns involved.

The introduced error estimation in the factorization was computed with the objective of verify the robustness of the LU factorization method. Each factorization is completely undone in every simplex method iteration. Thus, the operations are performed in the reverse order of the LU factorization and the matrix that will simulate the basis factorization update is obtained from the one being factored from the very first column. Therefore, the original basis is obtained with some amount of numerical error and the accumulated errors are a worst case estimation for each iteration.

4. Numerical Experiments

Initially, it was carried through numeric experiments for verifying the efficiency of the basis column update approaches presented in the previous section. Table 1 shows iterations (phase 2) and factorizations number of a subset of *Netlib* problems. Tables 2 and 3 show the nonzero number entries of the basis factorization of the same linear optimization problems using and not using the *Crash* basis, respectively.

In Table 4, the error estimate introduced by these operations was verified, computing the norm of the difference between the basis obtained by completely undoing the factorization and the original basis obtained directly from the constraint matrix.

It can be concluded that the factorization update proposed method is very robust. The maximum accumulated error in the worst case is of the order of 10^{-12}, that is, in this approach it is not necessary at all to refactorize the basis

LO Problems	Updates	Iterations	
		With Crash	Without Crash
kb2	3	55	82
adlittle	3	67	67
sc205	2	52	208
scsd1	8	314	223
bandm	3	297	197
scfxm1	2	138	173
beaconfd	1	26	27
scsd6	14	917	972

Table 1: Linear optimization problems data

nnz	Maximum			Minimum			Average		
	U_1	U_2	U_{MINOS}	U_1	U_2	U_{MINOS}	U_1	U_2	U_{MINOS}
kb2	343	460	406	174	210	130	260	333	257
adlittle	358	501	501	299	314	248	327	393	368
sc205	1099	1795	897	675	677	439	874	1109	647
scsd1	935	1275	907	495	577	371	674	881	602
bandm	5446	9669	4163	3221	3818	1977	4471	6411	3072
scfxm1	2164	4743	2059	1971	3396	1364	2092	4004	1691
beaconfd	1321	1846	1217	1296	1610	1194	1310	1794	1206
scsd6	2287	2886	1934	1168	1348	580	1728	1817	1257

Table 2: Nonzero entries number with Crash basis

any time, as it is usually done by the tradicional methods due to robustness and sparsity considerations.

5. Conclusions

In this work a static reordering of matrix columns is proposed, leading to simplex base with sparse LU factorizations and inexpensive factorization updates. The reordering has no initialization or updating costs since there is no need to reorder the columns in the factorization.

Stable results are present in Table 4. It is safe to conclude that no periodical factorization is needed for these problems as it is usually for updating schemes (Duff et all [3] and Maros [6]). As a result, if the starting basis is the identity, is not necessary to compute any LU factorization at all.

nnz	Maximum			Minimum			Average		
	U_1	U_2	U_{MINOS}	U_1	U_2	U_{MINOS}	U_1	U_2	U_{MINOS}
kb2	331	612	387	86	86	44	231	406	230
adlittle	353	697	528	267	363	262	324	521	374
sc205	1150	2091	1221	411	411	206	712	917	560
scsd1	1243	1216	1013	485	516	342	756	820	625
bandm	5803	22430	4173	3981	18493	2229	4812	20430	3121
scfxm1	2251	4718	2090	1780	3352	1279	2095	4227	1654
beaconfd	1471	3743	1485	1410	3571	1421	1443	3646	1458
scsd6	2528	3832	1975	1040	1434	615	1735	2424	1194

Table 3: Nonzero entries number without Crash basis

The U_1 basis update approach reduces up to 54% for the problem scfxm1 and 74% for the problem bandm nonzero entries, in comparison to the U_2 approach in the Tables 2 and 3, respectively. Thus, U_1 is used to carry through the LU factorization proposed in this work. The U_1 update approach is a little more dense that the U_{MINOS} update, but as we shall see it not need to refactorize the base. Thus, it probably will be faster than other approaches.

error	Maximum	Minimum	Average
kb2	2.7e-13	0	6.7e-14
adlittle	3.9e-15	1.7e-16	2.6e-15
scs205	1.8e-14	0	2.7e-15
scsd1	9.2e-15	4.4e-16	5.9e-15
bandm	1.9e-12	5.7e-14	1.1e-12
scfxm1	2.3e-13	2.6e-15	1.1e-13
beaconfd	3.1e-14	2.0e-14	2.7e-14
scsd6	3.2e-14	5.7e-16	2.0e-14

Table 4: Error estimate of updated basis without Crash basis

In spite of the U_1 basis update approach to obtain a basis be a little more dense that U_{MINOS} approach, it not refactor, thus it probably will obtain better computational time. For future work this approach will be integrated to implementation such as *MINOS* or *GLPK*.

Acknowledgments

This research was sponsored by the Foundation for the Support of Research of the State of São Paulo (FAPESP) and the Brazilian Council for the Development of Science and Technology (CNPq).

References

[1] R. Bartels, A stabilization of the simplex method, *Numer. Math.*, **16** (1969), 414-434.

[2] G. Bressan, A. Oliveira, Fast iterations for the combined cutting-stock and lot-sizing problems, In: *Annals of the IX International Conference on Industrial Enginnering and Operations Management*, Arq. TI0601 (2003), 1-8.

[3] I. Duff, A. Erisman, J. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford (1986).

[4] J.L. Kennington, R.V. Helgason, *Algorithms for Network Programming*, Wiley, New York (1980).

[5] H. Markowitz, The elimination form of the inverse and its applications to linear programming, *Management Science*, **3** (1957), 255-269.

[6] I. Maros, *Computational Techniques of the Simplex Method*, Kluwer Academic Publishers (2003).

[7] S.L. Nonas, A. Thorstenson, A combined cutting-stock and lot-sizing problem, *Operations Research*, **120** (2000), 327-342.

[8] J. Reid, A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming*, **24** (1982), 55-69.

[9] U. Suhl, L. Suhl, A fast LU update for linear programming, *Annals of Operations Research*, **43** (1993), 33-47.