On the sparsity issues of interior point methods for quadratic programming *

Csaba Mészáros

WP 98-4
Laboratory of Operations Research and Decision Systems
Hungarian Academy of Sciences
March, 1998

Abstract

In this paper we will investigate how the sparsity of non-separable quadratic programming problems behaves in interior point methods. We will show that for the normal equation approach, two orderings can be performed in independent steps to reduce the fill-in during interior point iterations. One of the permutations has to be performed on the columns, whereas the other on the rows of the problem. We point out that one can easily attribute the sparsity issues of non-separable quadratic programming problems to that of linear programming for which well–developed techniques are available. We describe how the fundamental structural properties of non–separable quadratic programming problems can be represented by a single matrix whose sparsity pattern can serve to determine the row permutation and to use heuristics developed for linear programming for determining which of the augmented system and normal equation approach is more advantageous. Numerical results are given on a wide variety of non–separable convex quadratic programming problems.

*This work was supported in part by EPSRC grant No. GR/J52655 and Hungarian Research Fund OTKA T-016413.

1 Introduction

The optimization practice showed that interior point methods (IPMs) are especially useful for solving large–scale problems. It was a common observation that the number of iterations, required by interior point methods, is virtually independent of problem size, thus the efficiency of these methods is closely related to the iteration speed. On large and sparse systems the iteration speed of IPMs is translated to the efficient exploitation of sparsity which is, in such a way, the perhaps most important issue of the large–scale optimization practice. Since linear programming is a natural and abundant source of very large and sparse problems, a significant effort was concentrated on the sparsity issues of linear programming with IPMs. These efforts manifested in a number of sophisticated techniques which make interior point methods very powerful for solving large and sparse linear programming problems [6, 1].

An important class of nonlinear optimization problems is the convex quadratic programming (QP) since the numerical techniques of QP are also the basis of successive quadratic programming methods in nonlinear optimization [5, 3]. Whereas the idea of using the separable equivalent of convex QP problems allows the straightforward application of the symbolic and numeric techniques of interior point methods for linear programming [2], numerical experiments indicated conclusively that the original non–separable formulation is more efficient for interior point methods in general [13].

The paper concerns the sparsity issues of IPMs for non–separable quadratic programming. We assume the quadratic programming problems in the form of

$$\begin{aligned} \min \quad & c^T x + \tfrac{1}{2} x^T Q x, \\ & Ax = b, \\ & x \geq 0, \end{aligned} \tag{1}$$

where $A \in R^{m \times n}$, $Q \in R^{n \times n}$ is symmetric positive semidefinite and $c, x \in R^n$, $b \in R^m$. In our investigation we will consider the case when Q and A are large and sparse matrices and the problem is non–separable, that is Q contains off–diagonal nonzero values. Figures 1 and 2 show the nonzero pattern of Q and A of the non–searable quadratic programming problem q25fv47.

The paper is organized as follows: In section 2 we describe the basic matrix operations of interior point methods. In section 3 we show some properties of the sparsity during the matrix computations described. We

Figure 1: Sparsity pattern, q25fv47 constraint matrix

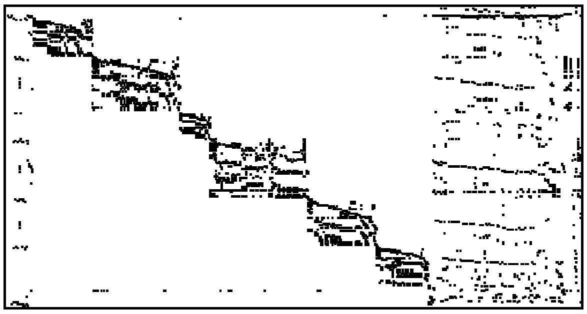
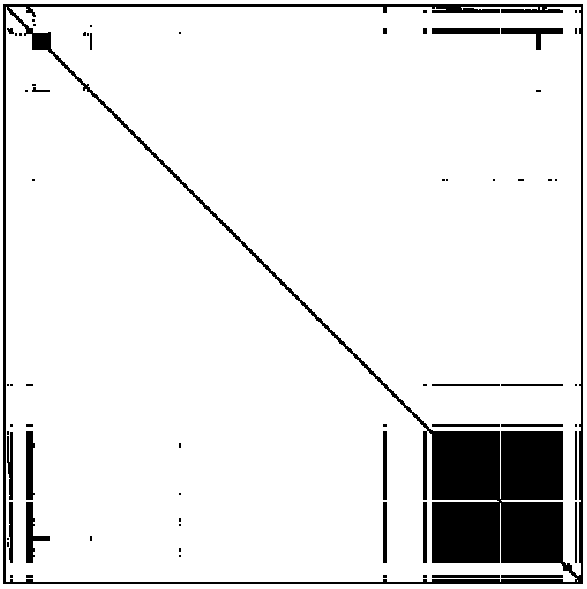


Figure 2: Sparsity pattern, q25fv47 quadratic objective



show that a single matrix can be formed whose sparsity pattern can serve for a structural analysis with techniques developed for linear programming. Computational results are given in section 4 on a set of non–separable convex quadratic programming problems. In section 5 we summarize our findings.

2 Matrix computations in IPMs for QP

Interior point methods are iterative approaches for solving optimization problems. During the past decade, a number of variants of interior point methods appeared in the literature, the reader should refer to [14, 16] for the full development of these algorithms. An important class of interior point methods are the log barrier algorithms, which perform especially efficiently in practice. A common property of these methods is that they require similar linear algebra operations during iterations. Since computationally the direct factorization approach is the most efficient, in the case of convex quadratic problems the effort to compute one iteration is mainly related to the cost to decompose the matrix of the *augmented system:*

$$\begin{bmatrix} -Q-D & A^T \\ A & 0 \end{bmatrix},$$

where $D \in R^{n \times n}$ diagonal matrix with positive diagonal values. In practice, symmetric permutation is used such that the fill–in during the decomposition is kept as low as possible. Thus the required decomposition can be written as

$$P \begin{bmatrix} -Q-D & A^T \\ A & 0 \end{bmatrix} P^T = L\hat{D}L^T, \tag{2}$$

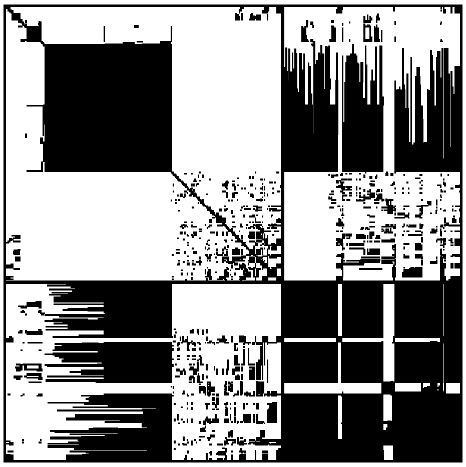
where $P \in R^{(n+m)\times(n+m)}$ is a permutation, $\hat{D} \in R^{(n+m)\times(n+m)}$ is diagonal and $L \in R^{(n+m)\times(n+m)}$ is lower triangular.

2.1 The normal equations approach

The normal equations approach uses two Cholesky factorizations to obtain the decomposition (2). By the first one, the negative definite $-Q-D$ block is eliminated first by performing the Cholesky decomposition

$$\hat{F}\hat{F}^T = P_1 (Q+D) P_1^T, \tag{3}$$

Figure 3: Sparsity pattern, factorization with NE on problem q25fv47



where $P_1 \in R^{n\times n}$ is a permutation and $\hat{F} \in R^{n\times n}$ is lower triangular. Then the remaining system, whose matrix is

$$A(\hat{F}\hat{F}^T)^{-1}A^T \tag{4}$$

positive definite, is factored by a second Cholesky factorization:

$$\hat{L}\hat{L}^T = P_2\left(A(\hat{F}\hat{F}^T)^{-1}A^T\right)P_2^T \tag{5}$$

where $P_2 \in R^{m\times m}$ is a permutation and $\hat{L} \in R^{m\times m}$ is lower triangular. In this way, the decomposition (2) can be written as:

$$L\hat{D}L^T = \begin{bmatrix} \hat{F} & \\ -A\hat{F}^{-T} & \hat{L} \end{bmatrix} \begin{bmatrix} -I & \\ & I \end{bmatrix} \begin{bmatrix} \hat{F}^T & -\hat{F}^{-1}A^T \\ & \hat{L}^T \end{bmatrix} \tag{6}$$

while P is the form of

$$P = \begin{bmatrix} P_1 & \\ & P_2 \end{bmatrix}.$$

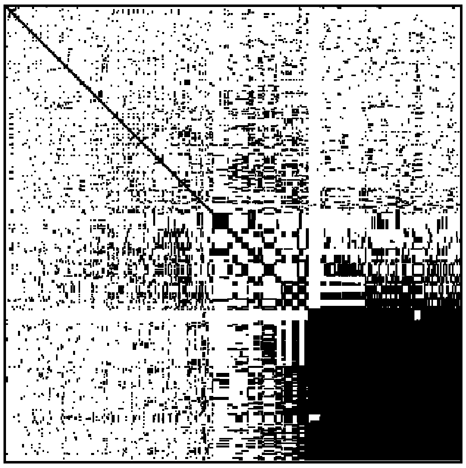
It is easy to see that permutation P_1 reorders the variables while P_2 reorders the constraints of the problem. Figure 3 shows the sparsity pattern of the decomposition (6) after reordering on problem q25fv47. Owing to the positive definite property, these permutations can be determined by pure symbolical investigations prior to applying the interior point method, without taking the numerical behavior into account. Since the sparsity pattern of $\hat{F}$ depends on the permutation used on $Q + D$, a natural question is how this permutation influences the pattern of $A(\hat{F}\hat{F}^T)^{-1}A^T$. We will show that the permutations for both of the Cholesky decompositions can be determined in independent steps because the structure of $A(\hat{F}\hat{F}^T)^{-1}A^T$ is unrelated to the permutation P_1.

2.2 The augmented system approach

As an alternative to the normal equation approach, one can use the direct symmetric factorization (2) without prescribing the elimination of the $-Q-D$ block at the beginning. In other words, by the augmented system approach it is not required that P is decomposable to an $n \times n$ and $m \times m$ block. Figure 4 shows the nonzero pattern of the factorization of the augmented system on problem q25fv47. As it was pointed out [8], this allows more freedom in the pivot sequence, which can be advantageous in several cases. It is to be noted that the matrix to be factored is indefinite, therefore, a technique other than the Cholesky factorization has to be used. One possibility to determine a proper pivot agenda is taking numerical properties into account when performing the factorization, as it was described in [10]. Alternatively, the pivot order can be determined by symbolic factorization if the matrix is quasidefinite [15]. Vanderbei suggested extending the primal problem with dummy slack variables with zero upper and lower bounds and using a trivial transformation on the augmented system, which makes the matrix quasidefinite. Another possibility is using dual regularization on the augmented system [11] which also provides the quasidefinite property.

It is commonly accepted now that both the augmented system and normal equations approaches have advantages and disadvantages and an important question is to determine which of them is the more beneficial for a given optimization problem. This decision has to be made at the very beginning of the interior point algorithm, prior to applying the decompositions. We will show that the analysis developed for linear programming [8] can be also applied succesfully in the quadratic programming practice.

Figure 4: Sparsity pattern, factorization with AS on problem q25fv47



3 On the sparsity issues of nonseparable QPs

In this section we will work with symbolic matrix transformations. This means that we will be interested in the possibly nonzero positions within a sparse matrix regardless of the numerical values. We will use the "$\equiv$" symbol to denote matrices with equvivalent sparsity structures. In this section we will denote nonzero entries in sparse matrices by the "$*$" symbol.

A natural expectation is that if A and Q are sparse and Q is such that F does not suffer heavy fill–in, furthermore, A is such that the Cholesky factorization of AA^T is sparse then the matrices arising during the computations by the normal equations approach are sparse and the normal equations approach is sufficiently efficient. Unfortunately, contrary to the expectations, heavy fill–in can occur even in the above case which can be demonstrated on the following simple example: Let Q and A be of the following structure:

$$Q \equiv \begin{bmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{bmatrix}, \qquad A \equiv \begin{bmatrix} * & & & \\ * & * & & \\ & * & * & \\ & & * & * \end{bmatrix}.$$

Clearly, the Cholesky factorizations of Q and AA^T suffer no fill–in that is matrices of these structures behave especially favourably. Since

$$F \equiv F^{-1} \equiv \begin{bmatrix} * & & & \\ * & * & & \\ & * & * & \\ & & * & * \end{bmatrix},$$

it is easy to see that $\hat{L}$ is completly dense because

$$A\left(FF^T\right)^{-1}A^T \equiv AF^TFA^T \equiv \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}.$$

Next let us consider the case when Q contains one off–diagonal nonzero value. Let

$$Q \equiv \begin{bmatrix} * & * & & \\ * & * & & \\ & & * & \\ & & & * \end{bmatrix} \qquad \text{and} \qquad A \equiv \begin{bmatrix} * & & & * \\ * & & & \\ & * & * & \\ & & * & * \end{bmatrix}.$$

After pivoting on $Q_{1,1}$ the sparsity pattern of the transformed augmented system will be

$$\left[\begin{array}{cccc|cccc} \odot & * & & & * & * & & \\ * & * & & & \otimes & \otimes & * & \\ & & * & & & & * & * \\ & & & * & * & & & * \\ \hline * & \otimes & & * & \otimes & \otimes & & \\ * & \otimes & & & \otimes & \otimes & & \\ & * & * & & & & & \\ & & * & * & & & & \end{array}\right],$$

where $\odot$ denotes the pivot position and $\otimes$ the fill–in entries. Let us observe that the pivot transformation resulted in a column in the lower off–diagonal block whose sparsity pattern is the union of the patterns of the first and second colums of A, corresponding to the column and row index of the off-diagonal entry in Q. Instead of forming $A\left(FF^T\right)^{-1}A^T$, one can use another matrix, namely

$$\hat{A} \equiv \begin{bmatrix} * & & * \\ * & & \\ * & * & \\ & * & * \end{bmatrix}$$

which has the property that $\hat{A}\hat{A}^T \equiv A\left(FF^T\right)^{-1}A^T$.

Definition 3.1 *Columns i and j are connected in Q if $i = j$ or $\exists\ i_1, ..., i_k$, such that $Q_{i_l,i_{l+1}} \neq 0$ for $l = 1, ..., k-1$ and $i_1 = i$, $i_k = j$.*

Definition 3.2 *$I_k \subseteq \{1, ..., n\}$, $k = 1, ..., l$ is an unconnected decomposition of Q if $\cup I_k = \{1, ..., n\}$, and if $i_1 \in I_i$ and $i_2 \in I_j$ are connected, then $i = j$. The set of columns defined by I_k is referred to as a connected partition.*

It is easy to see that the unconnected decomposition of a sparse matrix is unique up to a permutation. The unconnected decomposition can be obtained by Tarjan's algorithm which passes the nonzero entries of the marix at most once.

Theorem 3.1 *Let $\{I_k\}_1^l$ be an unconnected decomposition of Q, and $\hat{A}$ the $m \times l$ matrix for which the sparsity pattern of the k-th colum is the union of the patterns of columns $j \in I_k$ of A. Then $A\left(FF^T\right)^{-1}A^T \equiv \hat{A}\hat{A}^T$ for any F lower triangular matrix for which there exsists a $P \in R^{n\times n}$ permutation such that $FF^T \equiv PQP^T$.*

It follows from the uniqness of the unconnected decomposition that $\hat{A}$ is also unique. Clearly, $\hat{A}$ contains less or equal number of columns and nonzero entries than A and one can obtain it efficiently by an evident modification of Tarjan's method.

Proof. P be an arbitrary permutation on Q and let F_i denote the factorization pattern of PQP^T at the i-th step of the symbolic Gaussian elimination for $i = 0, ..., n$. Note that F_0 is the pattern of the lower triangular part of PQP^T and $F_n \equiv F$ where $FF^T \equiv PQP^T$. Clearly, for $i = 1, ..., n$ and $k \geq i$, $j > k$

$$(F_i)_{j,k} \neq 0 \iff \begin{cases} (F_{i-1})_{j,k} \neq 0 \quad \text{or} \\ (F_{i-1})_{j,i-1} \neq 0 \text{ and } (F_{i-1})_{k,i-1} \neq 0 \end{cases}. \tag{7}$$

Let us take a look at the pattern of $AF_i^T \equiv AF_i^{-T}$ which is formed during the elimination of the augmented system (2). At the i-th step of the elimination the column i of AF_{i-1}^T inherits ist structure to columns $j > i$ for which $(F_{i-1})_{j,i} \neq 0$. Taking the property (7) into account, one can see that the transformation effect in AF_i^Tcaused by a fill–in element in F_{i-1} can be traced back to the elements in F_0 that is to the original nonzero entries in Q. Let us observe that during such a traceback only the connected columns of Q can play role.

It follows from the above that at the elimination of the last column in the connected partition I_k the column in AF^T corresponding to the pivot will contain the sparsity pattern of all $j \in I_k$ columns of A. Thus when building the pattern of AF^TFA, all the columns of AF^T can be omitted except of the last pivot column in each connected partition. Clearly, the sparsity pattern of these columns is unique and these comlumns form $\hat{A}$. ■

The important consequence of the theorem is that the pattern of $A(FF^T)^{-1}A^T$ is independen of the P permutation and can be represented as the pattern of $\hat{A}\hat{A}^T$. This shows that the permutations on $Q + D$ and $A(FF^T)^{-1}A^T$ are unrelated and can be determined in independent steps. The significant advantages of working on $\hat{A}$ instead of working on the sparsity pattern of AF^T are that it does not require to determine F and since in general $\hat{A}$ contains less columns and nonzeros than AF^T, it is less difficult to build and requires less space to store.

Let us observe that building $\hat{A}$ allowes to use the sparsity heuristics developed for the interior point methods of LP. Perhaps the most important sparsity issue is to decide which of the augmented system and the normal equations approach is more beneficial for the present problem. By our approach this investigation on non-separable QP problems can be performed on $\hat{A}$ by the methods developed for linear programming [8].

Table 1: Problem statistics

Problem	Statistics on A			Off-diagonal Q
name	rows	columns	nonzeros	nonzeros
cvxqp1-l	5000	10000	14998	29984
cvxqp1-m	500	1000	1498	2984
cvxqp2-l	2500	10000	7499	29984

ctd. →

Problem	Statistics on A			Off-diagonal Q
name	rows	columns	nonzeros	nonzeros
cvxqp2-m	250	1000	749	2984
cvxqp3-l	7500	10000	22493	29984
cvxqp3-m	750	1000	2243	2984
dualc2	9	7	51	21
dualc5	1	8	8	28
dualc8	15	8	105	28
gouldqp2	349	699	1047	348
gouldqp3	349	699	1047	697
laser	1000	1002	3000	3000
mosarqp1	700	2500	3422	45
mosarqp2	600	900	2930	45
q25fv47	687	1456	10489	57674
qbandm	162	317	1421	5
qbeaconf	59	117	545	8
qbrandy	93	170	1582	27
qcapri	148	214	1448	824
qe226	149	250	2161	737
qetamacr	308	516	1898	2584
qfffff80	276	617	4649	1628
qforplan	89	362	5057	351
qgfrdxpn	432	908	2005	108
qgrow15	300	645	5620	462
qgrow22	440	946	8252	787
qgrow7	140	301	2612	327
qisrael	163	141	2256	656
qpilotno	731	1754	11833	163
qsc205	141	140	560	10
qscagr25	229	380	1296	89
qscagr7	60	103	319	17
qscfxm1	238	390	2195	674
qscfxm2	473	777	4433	1049

ctd. →

Problem	Statistics on A			Off-diagonal Q
name	rows	columns	nonzeros	nonzeros
qscfxm3	712	1168	6559	1109
qscorpio	160	203	513	13
qscrs8	221	905	2542	60
qscsd6	147	1350	4316	1308
qscsd8	397	2750	8584	2370
qsctap1	284	480	1638	117
qsctap2	1033	1880	6489	636
qsctap3	1408	2480	8595	861
qseba	55	110	343	433
qshare1b	94	204	996	21
qshell	293	1280	2517	2010
qship04l	288	1886	4267	42
qship04s	188	1238	2804	42
qship08l	478	3107	7628	33561
qship08s	256	1562	3820	8165
qship12l	638	4175	10513	43156
qship12s	323	1902	4777	8715
qsierra	1098	1912	6861	56
qstair	254	282	5087	703
qstandat	254	787	1707	666
stcqp1	0	3158	0	1525
stcqp2	0	2045	0	5898

4 Numerical results

In this section we shall compare the efficiency of the computations in interior point methods by the normal equations and augmented system approaches. Our tests were performed on the non–separable problems of the convex quadratic problem library [7] with BPMPD version 2.21 [12], which is an implementation of the primal–dual target following algorithm [4] for linear and convex quadratic programming problems. Our implementation is written in C and was compiled with the compiler system Watcom C/C++ version 10.6. The experiments were performed on an IBM PC compatible machine equipped with one Intel Pentium 200 MMX processor and 64 Mbyte of memory.

Table 1 contains the statistics of the test problems after the default presolve of BPMPD. Figures given include the number of rows, columns, nonzeros in A and the number of off–diagonal nonzeros in the lower triangular part of Q. For the statistics of these problems without presolve the reader should refer to [7].

Table 2 compares the number of off-diagonal nonzeros in L, the number of flops (in thousands) needed to compute one factorization (6) and the time (in seconds) needed to determine the ordering by the normal equations (NE) and augmented system (AS) approach, respectively. One flop is meant a pair of operations consisting of one multiplication and addition. Since the augmented system approach contains the normal equations approach as a special case, one can expect that the ordering performed on the augmented system does not generate results worse than the other. The experiments show that this is true except for two problems, qship08l and qship12l. We have experimented with different ordering algorithms, including the minimum local fill–in, minimum degree and approximated minimum degree heuristics, but on these problems the results obtained by the normal equations approach were better in all cases. We are convinced that the reason of this very unusual behavior is that ordering methods used are local pivot search heuristics which do not consider the global behavior. Note that on several problems (about 30 % of the cases) the augmented system approach generates significantly better factorization. The results also show that the overhead in the ordering time is not significant on problems where both approaches generated roughly the same ordering quality.

Table 2: Factorizations with the normal equation and augmented system approach

Problem	Nonzeros in L		Flops		Ordering time	
name	NE	AS	NE	AS	NE	AS
cvxqp1-l	*	3106765	*	154267	*	45.43
cvxqp1-m	196555	66521	3058	541	0.93	0.83
cvxqp2-l	*	2400979	*	112100	*	20.68
cvxqp2-m	77732	47190	608	297	0.28	0.45
cvxqp3-l	*	3694095	*	211356	*	80.07

ctd. →

Problem name	Nonzeros in L NE	AS	Flops NE	AS	Ordering time NE	AS
cvxqp3-m	373042	72540	8739	585	2.70	1.23
dualc2	116	72	0	0	0.03	0.02
dualc5	36	36	0	0	0.03	0.02
dualc8	251	133	0	0	0.03	0.02
gouldqp2	122847	2091	1429	0	0.45	0.03
gouldqp3	153820	3826	2145	1	0.49	0.05
laser	1005001	6000	33634	1	8.20	0.04
mosarqp1	21147	20851	29	28	0.12	0.11
mosarqp2	19856	19539	32	31	0.12	0.11
q25fv47	244300	95712	4053	775	1.11	1.06
qbandm	4209	3796	4	3	0.05	0.04
qbeaconf	1012	996	0	0	0.03	0.03
qbrandy	3494	3341	4	4	0.04	0.03
qcapri	10947	3261	40	3	0.05	0.04
qe226	6897	5989	11	8	0.04	0.05
qetamacr	22240	17518	49	47	0.10	0.18
qfffff80	21138	16484	46	35	0.10	0.18
qforplan	8304	7540	10	8	0.06	0.05
qgfrdxpn	4684	3297	3	1	0.04	0.03
qgrow15	12336	11830	13	11	0.17	0.17
qgrow22	18518	17614	20	17	0.25	0.25
qgrow7	5752	5495	6	5	0.08	0.08
qisrael	15873	4411	70	7	0.06	0.07
qpilotno	54808	53246	223	219	0.42	0.40
qsc205	2229	1912	2	1	0.03	0.03
qscagr25	3698	3285	2	2	0.04	0.03
qscagr7	863	793	0	0	0.03	0.02
qscfxm1	9150	5541	19	5	0.05	0.04
qscfxm2	15619	11389	27	10	0.08	0.06
qscfxm3	21233	17110	32	15	0.10	0.09
qscorpio	1384	1328	0	0	0.03	0.02

ctd. →

Problem	Nonzeros in L		Flops		Ordering time	
name	NE	AS	NE	AS	NE	AS
qscrs8	6297	6016	5	5	0.05	0.06
qscsd6	9052	8772	7	6	0.06	0.06
qscsd8	19782	18451	21	15	0.11	0.11
qsctap1	4188	4076	2	2	0.05	0.04
qsctap2	21723	20938	29	27	0.13	0.12
qsctap3	29447	29147	41	43	0.16	0.21
qseba	2274	874	3	0	0.04	0.03
qshare1b	2232	1944	1	1	0.03	0.03
qshell	9910	7711	13	7	0.08	0.08
qship04l	6857	6609	2	2	0.05	0.04
qship04s	4548	4300	1	1	0.04	0.03
qship08l	95199	116273	391	783	0.31	1.60
qship08s	28370	26164	65	62	0.11	0.42
qship12l	134103	168732	547	1030	0.44	1.45
qship12s	34491	26618	79	37	0.12	0.36
qsierra	16763	16503	11	11	0.14	0.08
qstair	22491	17084	90	53	0.10	0.12
qstandat	4968	4335	3	2	0.04	0.04
stcqp1	1525	1525	0	0	0.07	0.08
stcqp2	15675	15675	55	55	0.27	0.27

* Number of fill–in exceeded the available memory

Table 3 compares the time needed to compute one factorization (in seconds), the number of iterations needed by our solver to achieve 8 digits optimality and the total execution time (in seconds) of BPMPD. The results show clearly that differences in the quality of the factorization are translated to differences in the execution time.

Our solver performed a symbolical analysis [9] prior to the ordering method to select the normal equations or the augmented system approach. The execution time, corresponding to the decision of this heuristic is underlined in table 3. These results show that the heuristic successfully identified all cases where the augmented system is significantly more advantageous over the normal equations approach. Note that the "default" selection resulted in the less efficient approach only that problems (qship08l and qship12l) which behave very unusual by the present symbolical pivot search methods.

Table 3: Performance of BPMPD with the normal equation and augmented system approach

Problem	Factorization time		Iterations		Solution time	
name	NE	AS	NE	AS	NE	AS
cvxqp1-l	*	124.44	*	10	*	1459.16
cvxqp1-m	2.18	0.37	9	9	24.98	5.62
cvxqp2-l	*	89.64	*	10	*	1040.90
cvxqp2-m	0.45	0.21	10	10	6.47	3.74
cvxqp3-l	*	179.08	*	10	*	2163.34
cvxqp3-m	6.72	0.41	9	10	74.13	7.26
dualc2	0.01	0.01	8	8	0.21	0.20
dualc5	0.01	0.01	7	7	0.19	0.18
dualc8	0.01	0.01	7	7	0.25	0.25
gouldqp2	1.22	0.01	6	8	10.70	0.40
gouldqp3	2.06	0.01	7	7	18.32	0.43
laser	32.66	0.01	9	9	346.26	0.80
mosarqp1	0.02	0.02	8	8	1.21	1.21
mosarqp2	0.02	0.02	9	9	0.93	0.95
q25fv47	3.43	0.51	17	18	72.86	18.73
qbandm	0.01	0.01	18	18	0.58	0.55
qbeaconf	0.01	0.01	14	14	0.33	0.33
qbrandy	0.01	0.01	15	15	0.43	0.43
qcapri	0.03	0.01	21	24	1.38	0.67
qe226	0.01	0.01	16	16	0.64	0.61
qetamacr	0.04	0.04	16	16	1.74	1.81
qffffff80	0.04	0.03	20	20	2.15	2.05
qforplan	0.01	0.01	19	20	0.88	0.90
qgfrdxpn	0.01	0.01	16	16	0.82	0.78
qgrow15	0.01	0.01	19	19	1.38	1.38
qgrow22	0.02	0.02	22	22	2.21	2.21
qgrow7	0.01	0.01	20	20	0.74	0.74
qisrael	0.04	0.01	17	19	1.45	0.58
qpilotno	0.17	0.17	23	23	8.11	8.07

ctd. →

Problem name	Factorization time NE	AS	Iterations NE	AS	Solution time NE	AS
qsc205	0.01	0.01	16	16	0.35	0.33
qscagr25	0.01	0.01	14	14	0.52	0.52
qscagr7	0.01	0.01	13	13	0.27	0.26
qscfxm1	0.01	0.01	18	19	0.93	0.75
qscfxm2	0.02	0.01	23	23	1.77	1.51
qscfxm3	0.03	0.02	24	24	2.58	2.26
qscorpio	0.01	0.01	10	10	0.33	0.33
qscrs8	0.01	0.01	18	18	0.91	0.91
qscsd6	0.01	0.01	11	11	0.78	0.78
qscsd8	0.03	0.02	11	11	1.56	1.50
qsctap1	0.01	0.01	15	15	0.54	0.54
qsctap2	0.04	0.03	14	14	1.85	1.80
qsctap3	0.05	0.05	15	15	2.62	2.68
qseba	0.01	0.01	12	12	0.43	0.40
qshare1b	0.01	0.01	17	17	0.37	0.37
qshell	0.01	0.01	24	24	2.03	1.88
qship04l	0.01	0.01	10	10	0.81	0.82
qship04s	0.01	0.01	10	10	0.61	0.60
qship08l	0.41	0.58	10	9	7.52	10.23
qship08s	0.07	0.05	10	10	2.02	2.13
qship12l	0.52	0.74	12	11	11.07	15.39
qship12s	0.08	0.04	11	12	2.51	2.38
qsierra	0.02	0.01	15	15	1.82	1.82
qstair	0.06	0.04	14	14	1.83	1.41
qstandat	0.01	0.01	14	14	0.71	0.70
stcqp1	0.01	0.01	8	8	1.38	1.43
stcqp2	0.04	0.04	9	9	2.11	2.10

* Number of fill–in exceeded the available memory

5 Conclusions

In the paper we have shown that the sparsity issues of interior point methods for linearly constrained nonseparable (quadratic) optimization can be transformed into the sparsity considerations of IPMs for linear programming. The transformation described needs one pass through the nonzeros of Q and A and forms a new matrix which has the same number of rows, furthermore, less or equal number of columns and nonzeros as A and its structure can serve for performing further investigations by the methods developed for linear programming. We have shown that the augmented system approach is perhaps the most important technique of computations in IPMs for non–separable problems, since this performs significantly better than the normal equations approach in several cases. Let us note that in general the Shur–complement mechanism, which is considered as an alternative to the augmented system approach in the computational methods of IPMs for LP, cannot provide comparable results because the number of columns needed to be handled with it is large.

References

[1] E.D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu. Implementation of interior point methods for large scale linear programs. In T. Terlaky, editor, *Interior point methods of mathematical programming*, pages 189–252. Kluwer Academic Publishers, 1996.

[2] T.J. Carpenter, I.J. Lustig, J.M. Mulvey, and D.F. Shanno. Separable quadratic programming via a primal-dual interior point method and its use in a sequential procedure. *ORSA Journal on Comput.*, 5:182–191, 1993.

[3] D. den Hertog. *Interior point approach to linear, quadratic and convex programming: Algorithms and complexity.* Kluwer Academic Publishers, 1994.

[4] B. Jansen, C. Roos, T. Terlaky, and J. P. Vial. Primal–dual target following algorithms for linear programming. *Annals of Operations Research*, 62:197–231, 1996.

[5] F. Jarre. Interior-point methods for convex programming. *Applied Mathematics and Optimization*, 26:287–311, 1992.

[6] I. J. Lustig, R. E. Marsten, and D. F. Shanno. Interior point methods for linear programming: Computational state of the art. *ORSA Journal on Comput.*, 6(1):1–15, 1994.

[7] I. Maros and Cs. Mészáros. A repository of convex quadratic programming problems. Technical Report DOC 97/6, Imperial College, Department of Computing, London, 1997.

[8] I. Maros and Cs. Mészáros. The role of the augmented system in interior point methods. *European Journal on Operational Research*, 107(3):720–736, 1998.

[9] Cs. Mészáros. The "inexact" minimum local fill–in ordering algorithm. Working paper WP 95–7, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1995.

[10] Cs. Mészáros. The augmented variant of ipms in two–stage stochastic linear programming. *European Journal on Operational Research*, 101(2):317–327, 1997.

[11] Cs. Mészáros. On free variables in interior point methods. *Optimization Methods and Software*, 9:121–139, 1997.

[12] Cs. Mészáros. The BPMPD interior point solver for convex quadratic problems. Working paper WP 98–8, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[13] Cs. Mészáros. The separable and non–separable formulations of convex quadratic problems in interior point methods. Working paper WP 98–3, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1998.

[14] C Roos, T. Terlaky, and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach.* John Wiley & Sons, 1996.

[15] R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM Journal on Optim.*, 5(1):100–113, February 1995.

[16] S. J. Wright. *Primal–dual interior point methods.* SIAM, T57.74.W75, University Science Center, Philadelphia, 1997.