Bull. Nov. Comp. Center, Num. Anal., 12 (2003), 41–47

Integer quadratic programming programs*

E.A. Kotel'nikov, G.I. Zabinyako

This paper considers the software of the integer and the mixed-integer quadratic programming, which is based on the method of branches and boundaries with one-sided branching. Some examples of the solution of test problems are presented.

Introduction

The algorithm of the integer and the mixed-integer quadratic programming (IQP) is intended for the solution of problems of the following form:

$$\text{to minimize } f(x) = \frac{1}{2}(x, Qx) + (c, x) \tag{1}$$

under the constraints

$$Ax = b, \tag{2}$$

$$\alpha \leq x \leq \beta, \tag{3}$$

$$x_j \text{ are integers for } j \in J. \tag{4}$$

Here A is $m \times n$ matrix; Q is a symmetric positive semi-definite $l \times l$ matrix with $l \leq n$; $c, x, \alpha, \beta \in R^n$; $b \in R^m$, J is a list of integer variables.

The input data of the problem are represented in the MPS-format [1]. The column-wise sparse format [2] is used to store the matrices A, Q; Q only diagonal and sub-diagonal entries being stored.

The method of branches and boundaries with one-sided branching is implemented in the IQP programs [3]. The process of solution reduces to solving a sequence of estimation problems of the quadratic programming.

1. Solution of estimation problems

The difference of the estimation problems from the original problem (1)–(4) is in that condition (4) is absent and in each i-th estimation problem, conditions (3) are substituted for the conditions $\alpha^i \leq x \leq \beta^i$. The vectors α^i and β^i are formed by the rules of the method of branches and boundaries, which will be considered in the sequel.

For the solution of estimation problems, the reduced gradient method [1], where the variables x are subdivided into the basis x_B, the superbasis

*Supported by the Russian Foundation for Basic Research under Grant 01-07-90367.

x_S and the nonbasis x_N variables, is used. In the matrix A, B-basis $m \times m$, S-superbasis $m \times s$ and N-nonbasis $m \times (n - (m + s))$ matrices are distinguished, respectively. It is assumed that B is a non-singular matrix, the variables x_N taking their boundary values.

In the current subspace of the superbasis variables, minimization is done by the conjugate gradient method. At the k-th step of the method, the following is successively defined:

1. The vector of dual variables

$$y^k = (B^{-1})^T \nabla f(x_B^k);$$

2. The reduced gradient vector

$$h^k = \nabla f(x_S^k) - S^T y^k;$$

3. The vector of direction of the superbasis variables in a subspace

$$p_S^k = -h^k + \frac{\|h^k\|_2^2}{\|h^{k-1}\|_2^2} p_S^{k-1};$$

4. The vector of direction of the basis variables in a subspace

$$p_B^k = -B^{-1} S p_S^k;$$

5. The optimal step along p^k

$$\lambda_k = -\frac{(h^k, p_S^k)}{(p^k, Qp^k)},$$

where the vector p^k is composed of p_B^k, p_S^k, and $p_N^k = 0$. Let $x^{k+1} = x^k + \lambda p^k$, where $\lambda = \min\{\lambda_k, \lambda_{\max}\}$, and

$$\lambda_{\max} = \arg\max_{\lambda}\{\lambda : \ \alpha^i \leq x^k + \lambda p^k \leq \beta^i\}.$$

If $\lambda_k \geq \lambda_{\max}$, then according to the reduced gradient method, the content of superbasis and other variables is changed, and the conjugate gradient method starts in the new subspace. Otherwise, the process of the conjugate gradient method continues till the reduced gradient (with prescribed accuracy) becomes equal to zero.

To decrease the influence of computational errors on the process of the conjugate gradient method, the accuracy of solution to the systems of equations $B^T y^k = \nabla f_B(x_B^k)$, $Bp_B^k = Sp_S^k$ is controlled, and if necessary, their solutions are defined more precisely.

To avoid "recycling" of the conjugate gradient method in ill-conditioned problems, it is reasonable to make use of the technique proposed in [4] which is in checking – after a certain number of iterations – the fulfillment of the inequality: $\|h^k\|_2^2 \leq c(k)\|\bar{h}^k - h^k\|_2^2$, where the value $c(k)$ increases with k growing, h is a reduced gradient with direct calculation of $\nabla f(x_S^k)$, and in $\bar{h}^k$ the recurrently calculated $\nabla f(x_S^k)$ is used. When fulfilling the inequality, the process of the conjugate gradient method in a given superbasis subspace is completed.

Upon completion of optimization by the conjugate gradient method in a current superbasis subspace, based on the dual estimations of nonbasis columns, the possibility to form a new set of superbasis variables is checked. If there are no variables to be transferred from nonbasis to superbasis ones, the estimation problem is considered to be solved. Otherwise, the conjugate gradient method in the new superbasis subspace is used.

The specific features of the implementation of procedures of recurrent recalculation of matrices reverse to the basis ones and the reconstruction of inverse matrices, as well as programs of data preparation are considered in [5].

2. Algorithm of Integer Quadratic Programming

Estimations of the boundaries of the object function (1) with allowance for (4) are obtained from solutions of estimation problems of the quadratic programming (QP). The branching variable in the integer linear programming (ILP) is often chosen by means of penalties [1, 3]. In the considered algorithm IQP, an attempt is also made to use, where possible, the penalty estimations when choosing a branching variable.

First, let us consider a linear case. Let the value of the basis variable x_j for $j \in J$ in the optimal basis of the next estimation problem be given as $x_j = [x_j] + v_j$, where $[x_j]$ is the integer part of x_j. Let P_j^+ stand for the penalty for an increase of x_j by the value $1 - v_j$, and P_j^- – for a decrease of x_j by the value v_j.

We will present a penalty estimation algorithm using P_j^+, as an example. Numbers of non-basis variables will be assigned in the two lists: in I_x^- we put numbers of variables fixed at the lower boundary and in I_x^+ – at the upper one. If the variable x_j occupies the p-th position in the basis, then $z = e_p^T B^{-1}$ is calculated, where e_p is the p-th ort vector in R^m. Then

$$P_j^+ = \min\Big\{ \min_{q \in I_x^-,\, a_{pq}<0} \Big\{(1 - v_j)\Big(\frac{d_q}{-a_{pq}}\Big)\Big\},\ \min_{q \in I_x^+,\, a_{pq}>0} \Big\{(1 - v_j)\Big(\frac{-d_q}{a_{pq}}\Big)\Big\}\Big\},$$

where $a_{pq} = (z, A_q)$, A_q is the q-th non-basis column of the matrix A, d_q is its reduced estimator. If the q-th nonbasis variable must be integer, then

$(1-v_j)(\frac{d_q}{-\alpha_{pq}})$ or $(1-v_j)(\frac{-d_q}{\alpha_{pq}})$, which are less than $|d_q|$, are replaced by $|d_q|$. The initial values of P_j^- and P_j^+ are assumed to be equal to $+\infty$.

Let us replace the penalties for the quadratic function f by penalties for the linear function $g(x) = (\nabla f(x), x - x^*)$, where x^* is solution of the next estimation problem QP. As the initial function is convex, the penalty estimations for $g(x)$ will not exceed the penalties for $f(x)$. In the nonlinear case, when calculating penalties, it is necessary to additionally take into account superbasis variables – if they are present in the optimal solution of the estimation problem.

The values d_q for superbasis variables are equal to zero (d_q is the q-th component of the reduced gradient), and when dealing with the next j-th variables, penalties do not turn to zero only if $\alpha_{pq} = 0$. For any j-th superbasis variable $P_j^+ = P_j^- = 0$.

Let $P_{\max} = \max_j \max\{P_j^-, P_j^+\}$. In the ILP problems $P_{\max} = 0$ in those estimation problems which have not a single optimal basis. In the nonlinear case, the possibility to obtain $P_{\max} = 0$ increases at the account of superbasis variables. However numerical experiments have shown that the use of penalties in some ILP problems allows a considerable decrease of the time needed for the solution.

If $P_{\max} = 0$, then a basis or a superbasis variable x_j, whose value is the farthest from the integer, is chosen. In this case, for the time of defining a branching variable we assume $P_j^+ = 1 - v_j$ and $P_j^- = v_j$.

Schemes of the method of branches and boundaries with one-sided branching enable us to use a compact form of the lists of estimation problems. Let the variable x_j with the penalties P_j^- and P_j^+ be chosen for a branching at a certain level k. If $P_j^- \leq P_j^+$, then as the next estimation problem, we choose a problem corresponding to P_j^- and in the list of estimation problems (in the auxiliary array h) the information about an alternative problem should be stored. To this end, the following assignments are performed: $h(1,k) = j$; $h(2,k) = \beta_j^i$, where β_j^i is the upper boundary of the variable x_j in the i-th estimation problem at the previous level $(k-1)$; $h(3,k) = 1$ if $f^i + P_j^+ \geq r^i$, and $h(3,k) = 0$ otherwise. Here f^i is the optimal value of f in the i-th estimation problem, and r^i is the current value of the incumbent; $h(4,k) = f^i$; $h(5,k) = P_j^+$. For $P_j^+ < P_j^-$ we choose an estimation problem corresponding to the penalty P_j^+ and the array h stores the values: $h(1,k) = -j$; $h(2,k) = \alpha_j^i$; $h(3,k)$ is equal to 0 or 1 depending on fulfillment of the inequality $f^i + P_j^- \geq r^i$; $h(4,k) = f^i$; $h(5,k) = P_j^-$.

Algorithm:

Step 0. Set $i = 0$, $k = 0$, the incumbent value $r^0 = +\infty$, $\alpha^0 = \alpha$, and $\beta^0 = \beta$.

Step 1. Solve the current QP problem. Let x^i be its optimal solution and $f^i = f(x^i)$.

a) if $f^i \geq r^i$ or the system of constraints is inconsistent, then assign $r^{i+1} = r^i$, $i = i + 1$, and go to Step 3;

b) let $f^i < r^i$. If the vector x^i is not integer, then go to Step 2. If the solution x^i is integer, then assign $r^{i+1} = f^i$, $i = i + 1$. For $f^i = f^0$ go to Step 5, else to Step 3.

Step 2. For basis and superbasis variables x_j, $j \in J$, whose x_j^i is not integer, calculate the penalties P_j^+, O_j^-. Among them find the minimum penalty $P_{\min}$.

a) if $f^i + P_{\min} \geq r^i$, then go to Step 3;

b) for $f^i + P_{\min} < r^i$, perform branching by the basis or superbasis variable x_j that corresponds to the maximum penalty P_j^+ or P_j^-. Choose the least from P_j^+ and P_j^-. If $P_j^- \leq P_j^+$, then choose the next estimation problem corresponding to P_j^-. Increase $k = k + 1$, put on the list the problem corresponding to the penalty P_j^+. Change the upper boundary of the variable x_j, setting it equal to $[x_j^i]$. Go to Step 1.

(If $P_j^+ < P_j^-$, then the list h stores the problem corresponding to P_j^-, and in the problem in question the lower boundary of the variable x_j is set equal to $[x_j^i] + 1$).

Step 3. If $k = 0$, then go to Step 5. For $k > 0$, test either $h(3, k) = 1$ or $h(4, k) + h(5, k) \geq r^i$, then go to Step 4. Otherwise, using the array h, formulate a problem alternative to that formed at Step 2.b. Assign $h(3, k) = 1$ and go to Step 1.

Step 4. Set two-sided constraints on the variable x_j, using the values $h(1, k)$ and $h(2, k)$. Assign $k = k - 1$ and go to Step 3.

Step 5. Stop.

Schemes of branches and boundaries with one-sided branching enable us with lower computer costs to perform the transition to the next estimation problem. The discussed realization of the scheme is similar in many respects to the realization for the ILP in [5].

3. Solution of problems

The test problems are borrowed from [6]. Table 1 lists the input parameters of problems. Here No is number of a problem; m is the total number of

Table 1

No	The name of a problem	m	m_e	n	n_i	n_b	nz
1	ibell3a	106	—	122	60	31	304
2	ibell5	91	—	104	58	30	266
3	idcmulti	290	78	548	75	75	1315
4	igesa3	1368	48	1152	384	216	4944
5	igesa3_0	1224	120	1152	672	336	3622
6	imas284	68	—	151	150	150	9631
7	imisco7	212	35	260	259	259	8619
8	iran13	195	26	338	169	169	676
9	iran8	296	40	512	256	256	1024
10	ivalues	1	1	202	202	—	202

constraints among which m_e constraints are equalities, n is the number of variables among which n_i are integer and n_b variables are Boolean; nz is the number of non-zeroes in the matrix A.

Problems 1 and 6–10 were taken from the IQP tests in [6]. In all these problems only the integer variables (x_j, $j \in J$) are nonlinear and in all the rest variables the object function is linear. The matrices Q are well conditioned. In these problems, except for problem 10, Q is diagonal, and in problem 10 Q contains 7240 non-diagonal entries. Problems 2–5 were obtained from the IQP-tests in [6] by adding to the object functions of quadratic terms in the variables x_j, $j \in J$.

Table 2 contains characteristics of the solution process of problems by the algorithm on the computer system MVS 1000M (processor ALPHA-21264 with the clock frequency 830 MHz). Here No is number of a problem; it is the total number of iterations in estimation problems; cp is the number of estimation problems; t is the time of solution in seconds; P_1 shows how many times the condition $f^i + P_{\min} \geq r^i$ was fulfilled in the process of problem

Table 2

No	it	cp	t	P_1	P_2	P_0
1	354065	35765	66.76	2680	9648	18
2	14952358	5263409	2311	279439	1593369	0
3	587901	20128	268.4	1407	3726	599
4	278756	8003	723.8	29	2456	0
5	508628	15646	1240	9	5516	0
6	1054772	53052	380.8	409	16672	127
7	3848924	54826	305.7	202	6797	2086
8	18496522	1324478	4480	8847	537387	6604
9	16239520	985215	5751	4385	400440	15115
10	279864	15955	31.43	0	0	7977

solution; P_2 – the number of cases when the condition $f^i + P_j \geq r^i$ was fulfilled, where $P_j = P_j^+$ or $P_j = P_j^-$; P_0 shows how many times $P_{\max} = 0$.

References

[1] Murtaf B. Modern Linear Programming. Theory and Practice. – Moscow: Mir, 1984.

[2] Pissanetsky S. Technology of Disperse Matrices. – Moscow: Mir, 1988.

[3] Kovalev M.M. Discrete Optimization (Integer Programming). – Minsk: Belarus University, 1977.

[4] Wilkinson J.H., Reinsch C. Manual of Algorithms in the Language ALGOL. Linear algebra. – Moscow: Mashinostroenie, 1976.

[5] Zabinyako G.I., Kotel'nikov E.A. Linear optimization programs // NCC Bulletin. Series Num. Anal. – Novosibirsk: NCC Publisher, 2002. – Issue 11. – P. 103–112.

[6] http://plato.la.asu.edu