Projected Barzilai-Borwein Methods for Large-Scale Box-Constrained Quadratic Programming *

Yu-Hong Dai † Roger Fletcher ‡

September 2, 2003

Numerical Analysis Report NA/215

Abstract

This paper studies projected Barzilai-Borwein (PBB) methods for large-scale box-constrained quadratic programming. Recent work on this method has modified the PBB method by incorporating the Grippo-Lampariello-Lucidi (GLL) nonmonotone line search, so as to enable global convergence to be proved. We show by many numerical experiments that the performance of the PBB method deteriorates if the GLL line search is used. We have therefore considered the question of whether the unmodified method is globally convergent, which we show not to be the case, by exhibiting a counter example in which the method cycles. A new projected gradient method (PABB) is then considered that alternately uses the two Barzilai-Borwein steplengths. We also give an example in which this method may cycle, although its practical performance is seen to be superior to the PBB method. With the aim of both ensuring global convergence and preserving the good numerical performance of the unmodified methods, we examine other recent work on nonmonotone line searches, and propose a new adaptive variant with some attractive features. Further numerical experiments show that the PABB method with the adaptive line search is the best BB-like method in the positive definite case, and it compares reasonably well against the GPCG algorithm of Moré and Toraldo. In the indefinite case, the PBB method with the adaptive line search is shown on some examples to find local minima with better solution values, and hence may be preferred for this reason.

*This work was supported by the EPRSC in UK (no. GR/R87208/01) and the Chinese NSF grant (no. 10171104).

†State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering computing, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, PO Box 2719, Beijing 100080, PR China. Email: dyh@lsec.cc.ac.cn

‡Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK. Email: fletcher@maths.dundee.ac.uk

1. Introduction

In this paper the box-constrained quadratic programming (BQP) problem

$$\begin{array}{ll} \min & q(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{b}^T\mathbf{x} \\ s.t. & \quad \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{array} \tag{1.1}$$

is studied, where $A \in \Re^{n\times n}$ is symmetric but may be indefinite, and $\mathbf{b}$, $\mathbf{l}$, $\mathbf{u}$ (with $\mathbf{l} \le \mathbf{u}$) are vectors in $\Re^n$. This problem not only is of practical interest in itself (many applications have been pointed out by Moré and Toraldo [21, 22] and Friedlander and Martínez [14]), but also arises as the trust region box-constrained subproblem of unconstrained optimization. In addition, the symmetric linear complementarity problem and the strictly convex linearly constrained quadratic program can be converted into the form (1.1). Once the optimal face of the box is identified, (1.1) reduces to the unconstrained minimization of a quadratic function. This means that an algorithm for (1.1) has two aspects: (a) to identify the optimal face and (b) to minimize the function on the face. However, the number of the faces of the box in (1.1) is combinatorial with n. A major difficulty for solving (1.1) is that of how to identify the optimal face when the dimension n is large.

Active set methods are a class of efficient algorithms for (1.1) if n is small. They usually restrict the change in the dimension of the working face by only dropping or adding one constraint at each iteration. This can be a serious disadvantage in the large-scale case, where many iterations may be needed to identify the optimal face if the initial active set and the optimal active set are significantly different. Another disadvantage is that the methods may require an accurate solution of the working face, which is inefficient if the current face is far from the optimal face. In addition, most active set methods use some kind of matrix factorization, which may be expensive if n is large.

Projected gradient (PG) methods provide an alternative way of solving large-scale BQP problems. They have the advantage that many constraints can be added or deleted from the working set on each iteration. They are also simple and easy to code, and avoid the need for a matrix factorization. Early references on PG methods can be dated back to Goldstein [17] and Levitin and Polyak [20], where constant steplengths are used. A modified Armijo line search is introduced by Bertsekas [3] for the choice of steplength. However, these early PG methods are often inefficient since their performance resembles the steepest descent method, which is usually very slow once the optimal face is identified. Nevertheless, as we will see, the effectiveness of projected gradient methods can be significantly improved by incorporating new and fast gradient methods for unconstrained optimization.

An alternative development for improving the effectiveness of projected gradient methods has been to make use of the conjugate gradient (CG) method for finding the minimizer on a face. There have been many conjugate gradient type projection algorithms for the positive definite case. Polyak [23] first proposes an algorithm that uses the negative projected gradient to leave a face and conjugate gradients to explore a face. Dembo and Tulowitzki [12] propose algorithms that can drop and add many constraints from the working face at each iteration, and do not require the accurate solution of the working face. Subsequently, Yang and Tolle [29] and Wright [28] have proposed two algorithms that are able to drop and add many constraints on each iteration, and they show that the algorithms terminate in a finite number of iterations. Although the analysis presented by Yang and Tolle is elegant, their algorithm may be inefficient in practice since it requires subproblems to be solved exactly. Wright [28] proposes to use the PG method until the binding constraints stay constant on two successive iterations, after which the CG method is used to explore the current face. Moré and Toraldo [22] argue that the PG phase of Wright's algorithm can require many iterations to identify a suitable face if the starting point is far from the solution. A new algorithm named GPCG is proposed in [22] that uses the PG method until either a suitable face is identified, or the PG method fails to make reasonable progress. At this stage the current face is explored by using the CG method. Once the CG method fails to make significant progress, a decision is made whether or not to switch back to the PG method. This strategy avoids the excessive exploration of non-optimal faces by the CG method. For the GPCG algorithm, Moré and Toraldo report good numerical results for real problems of up to 15625 variables, and show an improvement over the algorithms of Dembo and Tulowitzki (see [22]). More recently, Friedlander and Martinéz [14] have developed a new method that decides to leave or stay the current face by the so-called "chopped gradient" and they report numerical results comparable to the algorithm in [22]. In this paper we use the GPCG algorithm as a standard against which to compare our new PG methods.

In this paper, we are interested in PG-type methods for solving (1.1), but with a steplength choice influenced by recent work on gradient methods for unconstrained optimization. We define Ω to be the feasible set of (1.1)

$$\Omega = \{\mathbf{x} \in \Re^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}, \tag{1.2}$$

and let P denote the projection operator on to Ω, that is

$$P(\mathbf{x}) = \mathrm{mid}(\mathbf{l}, \mathbf{u}, \mathbf{x}), \tag{1.3}$$

where $\mathrm{mid}(\mathbf{l}, \mathbf{u}, \mathbf{x})$ is the vector whose i-th component is the median of the set $\{l_i, u_i, x_i\}$. Assuming the current point $\mathbf{x}_k$ that is feasible and denoting $\mathbf{g}_k = A\mathbf{x}_k - \mathbf{b}$, the projected gradient method computes the next point by

$$\mathbf{x}_{k+1} = P[\mathbf{x}_k - \alpha_k \mathbf{g}_k], \tag{1.4}$$

where $\alpha_k > 0$ is some steplength.

For the minimization of a strictly convex quadratic $q(\mathbf{x})$ without constraints, the classical (Cauchy) steepest descent (SD) method calculates the steplength from

$$\alpha_k^{SD} = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T A \mathbf{g}_k}. \tag{1.5}$$

This choice of steplength minimizes $q(\mathbf{x})$ along the steepest descent direction. The SD method is known to be Q-linearly convergent, but the rate of convergence can be very slow if the matrix A is badly conditioned (see Akaike [1]), so much so that the method will often fail to solve a problem in a reasonable time.

A significant development that has completely changed our perspectives on the effectiveness of gradient methods is due to Barzilai and Borwein [2]. They propose two choices of the steplength

$$\alpha_k^{BB1} = \frac{\mathbf{s}_{k-1}^T \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}} \tag{1.6}$$

and

$$\alpha_k^{BB2} = \frac{\mathbf{s}_{k-1}^T \mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^T \mathbf{y}_{k-1}}, \tag{1.7}$$

where $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$. A feature of these methods is that they are non-monotonic, that is $q(\mathbf{x}_k)$ may increase on some iterations, in contrast to the SD method. Barzilai and Borwein prove an R-superlinear convergence result for the new formulae, when applied for 2-dimensional strictly convex quadratics. Also Raydan [24] has shown that the BB method (with either steplength formula) is globally convergent in the strictly convex quadratic case. A recent study of Dai and Fletcher [8] has shown that the BB method with either (1.6), (1.7) or any alternate use of the formulae is likely to be asymptotically R-superlinearly convergent in the 3-dimensional case, but not for $n > 3$. In general, R-linear convergence has been established for the BB method by Dai and Liao [9] in the $n-$dimensional case. Numerical experiments have shown that the BB method is far faster than the SD method. Moreover, Fletcher [13] has shown that the BB method is not greatly inferior to the CG method for a convex quadratic with 10^6 variables. By incorporating the nonmonotone line search of Grippo, Lampariello and Lucidi [18], Raydan [25] has extended the BB method with the choice (1.6) for solving general unconstrained optimization. A wide range of numerical experience is reported in [25] on problems of up to 10^4 variables, showing that the method compares reasonably well against the Polak-Ribière and CONMIN conjugate gradient techniques. Birgin, Martínez and Raydan [4] have extended the work of Raydan and have proposed efficient projected BB algorithms (SPG1 and SPG2) for the minimization of differentiable functions on closed convex sets. The BB method has now received many generalizations and applications, see Birgin, Martínez and Raydan [5], Dai and Fletcher [8], Serafini, Zanghirati and Zanni [26] and the references therein. Aside from the BB formulae, there have also been some other steplength choices which have been shown to work well, and we consider these later in the section.

The success of these new gradient methods motivates us to re-consider the projected gradient method (1.4) for BQP problems. The fact that the BB method is efficient in the exploration of a face indicates there is no need to make a decision as to whether to leave the current face or to explore it further. However there are several variants of the BB method, and we first investigate which performs best in the context of a projected gradient method. For unconstrained optimization the formula (1.6) is generally believed to outperform (1.7), but we shall see that this does not seem to be the case here. There are also other options to evaluate. We refer to the projected gradient method (1.4) with the steplength given by (1.6) as the projected BB method or PBB method. For reasons described in §2, we have also been motivated to consider the alternate use of the BB formulae (1.6) and (1.7)

$$\alpha_k^{ABB} = \begin{cases} \alpha_k^{BB1}, & \text{for odd } k; \\ \alpha_k^{BB2}, & \text{for even } k. \end{cases} \tag{1.8}$$

which we refer to as the projected alternating BB or PABB method. We note that alternating strategies involving the formulae (1.6), (1.7), or modified BB steplengths have been studied in Dai, Yuan and Yuan [10], Grippo and Sciandrone [19], and Zanghirati and his cooperators [16, 30, 26], and that the formula (1.8) has already been used by Grippo and Sciandrone [19] in the context of unconstrained optimization. More details of these alternating strategies are given in §2. Our numerical tests in §2 on random positive definite BQP problems show that the PABB method without line searches is usually successful and performs more efficiently than the PBB method.

We also note that the steplength (1.6) may be written as

$$\alpha_k = \frac{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}{\mathbf{g}_{k-1}^T A \mathbf{g}_{k-1}},$$

which is the SD steplength at the previous point $\mathbf{x}_{k-1}$. We express this fact by saying that the BB steplength (1.6) uses the SD steplength with one delay. A class of BB-like methods with longer delays (retards) has been investigated by Friedlander *et al* [15] in the unconstrained case. We have also used one family of gradient methods from [15] (see (2.2) for the family) in §2 to study the influence of delays in the calculation of the BB-like steplengths in the box-constrained case. Our experiments show that there is no advantage in using more than one delay in the box constrained case.

Although both the PBB and PABB methods have never failed in our existing tests on BQP problems, even without line searches, we have been able to construct strictly convex BQP problems in two variables showing that the unmodified PBB and PABB methods may cycle between several points, and hence fail to converge. These counter-examples are given in §3. This indicates that it is not possible to extend the convergence results of the BB method for positive definite linear systems given in [24], to BQP problems, even if they are strictly convex. The introduction of some kind of line search is therefore necessary to ensure the global convergence. As mentioned before, the Grippo-Lampariello-Lucidi (GLL) nonmonotone line search has been incorporated with the projected BB method in [4], for the minimization of differentiable functions on closed convex sets. However, as will be shown in §4, the GLL nonmonotone line search may significantly degrade the performance of the

PBB and PABB methods. In fact, numerical results of the two methods without line searches are often better than those with the GLL line search. Therefore in §4 we suggest a simplified adaptive nonmonotone line search based on the work of Toint [27] and Dai and Zhang [11]. An important difference between our method and the adaptive techniques in [27] and [11] is that in our work the initial reference function value is set to be $+\infty$. We find that this line search not only guarantees global convergence of the PBB or PABB method, but also is almost always able to accept the steplength of the unmodified method, and hence is able to preserve its good numerical performance.

It is also important to consider how the PBB and PABB methods compare with PG methods that incorporate CG steps. For reasons described in §1, we take the GPCG method of Moré and Toraldo [22] as our standard of comparison. Further numerical experiments in §5 on large-scale positive definite BQP problems show that the PABB method with an adaptive nonmonotone line search is again the best variant amongst the BB-like projected gradient algorithms considered in this paper, and it also compares reasonably well against the GPCG algorithm. However, for indefinite BQP problems, we find that the PBB method with an adaptive nonmonotone line search can often find a better solution point than the GPCG algorithm and the other variants of projected gradient algorithms, and hence is recommended in this case. Further discussion is given in §6.

2. Projected gradient methods without line searches

Throughout this section we assume that the Hessian matrix A is symmetric and positive definite (SPD). Our first experiments concern the effectiveness of the formulae (1.6) and (1.7). We shall see in §3 that it is possible for the projected gradient method to fail if these formulae are used, in contrast to the unconstrained case for which a global convergence proof [24] exists. However our results indicate that failure would appear to be very unlikely in practice. We have generated 15 random box consrained SPD test problems with $n = 1000$. The upper and lower bounds are chosen so that a range of different problems types is generated, with the number of active constraints at the solution ranging from 14 to 885. Also a range of condition numbers from 10^4 to 10^6 is used. More details on how the problems are generated is given in §5. The stopping condition is

$$\|\nabla q_\Omega(\mathbf{x}_k)\|_2 \leq 10^{-5}\|\mathbf{g}_1\|_2. \tag{2.1}$$

The results are shown in Table 1, where 10^{ncond} is the condition number of the Hessian and $na(\mathbf{x}_1)$, $na(\mathbf{x}^*)$ denote the number of initial active constraints and active constraints at the optimal solution $\mathbf{x}^*$, respectively.

First we consider the PBB method that uses formula (1.6), and we see that it is very effective in solving all the problems, with an average of 127 iterations, and never more than 218 iterations. The number of iterations is seen to be somewhat greater than would be needed to solve the unconstrained problem, but not greatly

Problem			PBB				PABB			
$ncond$	$na(\mathbf{x}_0)$	$na(\mathbf{x}^*)$	it_U	it	it_f	it_F	it_U	it	it_f	it_F
4	679	39	47	61	29	34	49	59	22	33
6	117	508	113	195	75	141	115	184	46	57
6	257	595	115	179	60	115	112	171	46	46
5	303	801	72	102	33	49	77	108	31	31
5	820	149	65	116	25	89	93	108	26	52
6	91	565	105	189	65	104	117	184	47	47
5	660	885	76	91	30	48	72	98	32	32
4	874	124	52	58	16	30	58	71	16	22
4	659	746	52	62	18	18	55	60	20	20
5	343	107	83	117	21	52	76	93	26	52
6	390	535	142	218	62	149	110	161	51	73
6	176	549	128	174	75	97	107	176	57	91
6	481	14	137	176	43	92	119	162	30	98
4	35	307	51	59	13	27	48	63	15	23
5	205	196	75	115	27	54	76	113	28	39
average			87.5	127	39	73	85.6	121	33	48

Table 1. Numerical comparison of BB-like methods

so. However, in examining the results in more detail, we observe that, whilst the correct active set is often located at an early stage in the calculation, it often can take many more iterations before the active set stays constant for all subsequent iterations. To see this, we have tabulated two quantities it_f and it_F. Here it_f is the first iteration on which the correct active set $\mathcal{A}(\mathbf{x}^*)$ is located, and it_F is the first iteration on which active set is identical to $\mathcal{A}(\mathbf{x}^*)$ for all subsequent iterations. It can be seen from Table 1 that typically it_F is twice as large as it_f for the PBB method.

We have also tested the projected gradient method that uses formula (1.7), but find that the resulting method is somewhat inferior to the PBB method. However, the alternating use of (1.6) and (1.7) in the PABB method gives better results, as also shown in Table 1. In particular, the correct active set can be seen to settle down more quickly, resulting in a modest but worthwhile improvement in the total number of iterations. An explanation of this behaviour may be due to the fact that the steplength (1.7) is smaller than (or possibly equal to) that in (1.6), which follows from the Cauchy-Schwarz inequality. Use of the larger steplength (1.6) may cause the current face to be quitted more often, which may cause slower convergence, and thus mitigate the advantage that (1.6) seems to have in the unconstrained case. However the use of (1.7) on every iteration seems to slow down the initial identification of $\mathcal{A}(\mathbf{x}^*)$, and so is not advantageous. For example, using a set of 50 random test problems with $n = 1000$, generated as for Table 1, we show in Table 2 the iteration number at which the correct active set is first identified. (Only those results where the difference is greater than 5 are tabulated.) Formula (1.6) wins in 16 of these 21 cases, and on average the margin of improvement is significantly larger.

(1.6):	30	49	34	31	13	22	62	76	14	25	56
(1.7):	36	72	42	40	23	15	55	82	59	35	49
(1.6):	68	15	32	28	56	55	73	35	47	60	
(1.7):	57	21	21	40	76	62	104	45	58	99	

Table 2. Effectiveness of (1.6) and (1.7) in locating the correct active set

The idea of alternately using the formula (1.6), (1.7) or modified BB steplengths has appeared in several references. The alternate use of modified BB steplengths and the formula (1.8) are currently introduced in [10] and [19] in the context of unconstrained optimization. [16] and [30] report good numerical results with variable projection methods that switch between (1.6) and (1.7) every three iterations and use a limited minimization rule . In addition, Serafini, Zanghirati and Zanni [26] suggest several criteria for the switching mechanism, and again report better numerical results with the variable projection methods. On the other hand, they find that the alternating strategy used in their methods is not suited to the SPG method in [4]. However, we can see from the above and §4 that the PABB method is somewhat better than PBB method, at least in the positive definite case. In addition, good numerical results, but not as good as the PABB method, have been obtained by switching between (1.6) and (1.7) every m_{switch} iterations for $m_{switch} = 2, 3$. It still remains to study whether there exists a more efficient alternating strategy similar to the one in [26] when a nonmonotone line search is used.

We have also considered other BB-like methods, in particular the family of gradient methods with maximal retards [15], in which the steplength is defined by

$$\alpha_k = \frac{\mathbf{s}_{k-\bar{m}}^T \mathbf{s}_{k-\bar{m}}}{\mathbf{s}_{k-\bar{m}}^T \mathbf{y}_{k-\bar{m}}}, \tag{2.2}$$

where $\bar{m} = \min(m, k-1)$ and $m \geq 1$ is some prefixed integer. We compare a range of values of m with the PBB method ($m = 1$) and the steepest descent method. For the steepest descent method, we use the steplength

$$\alpha_k = \frac{\|q_\Omega(\mathbf{x}_k)\|_2^2}{q_\Omega(\mathbf{x}_k)^T A q_\Omega(\mathbf{x}_k)}, \tag{2.3}$$

where

$$\nabla[q_\Omega(\mathbf{x}_k)]_i = \begin{cases} (\mathbf{g}_k)_i & \text{if } x_i \in (l_i, u_i) \\ \min\{(\mathbf{g}_k)_i, 0\} & \text{if } x_i = l_i \\ \max\{(\mathbf{g}_k)_i, 0\} & \text{if } x_i = u_i. \end{cases} \tag{2.4}$$

For all these methods, a line search is only carried out on the first iteration. Using a set of 50 random test problems with $n = 1000$, generated as for Table 1, we show in

Table 3 the number of the problems that are successfully solved in 2000 iterations by each method.

SD	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
18	50	50	26	7	2

Table 3. Successful solutions for 50 random SPD problems

We see that performance deteriorates significantly as m is increased, and even for $m = 2$ it happens that the numerical results are almost uniformly worse than those for the PBB method.

We have also made some experiments using the AS and CSDS methods (see [5]) in the BQP context. However we have found that these do not compete with the PABB method described here. A possible reason may be that repeatedly changing the active set destroys the theoretical basis (see [8]) that underpins these methods.

3. Two-dimensional counter-examples

In this section, we describe in detail a 2-dimensional counter-example which shows that the PBB method without line searches may cycle between five points. A counter-example with eight cyclic points is also given for the PABB method.

We consider the following box-constrained quadratic programming problem

$$\begin{array}{ll} \min & \frac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} t+1 & t-1 \\ t-1 & t+1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ s.t. & \qquad x_1 \geq -3, \quad x_2 \geq 1, \end{array}$$

where $t \geq 31$ is some constant. The unconstrained minimizer of the quadratic function is $\mathbf{x}_u^* = (0,0)^T$. By the Karush-Kuhn-Tucker conditions, the optimal solution of the problem is $\mathbf{x}^* = (-\frac{t-1}{t+1}, 1)^T$. We take the initial point and steplength to be

$$\mathbf{x}_1 = \begin{pmatrix} -3 \\ 1 \end{pmatrix} \quad \text{and} \quad \alpha_1 = \frac{1}{t+1}.$$

For any $\mathbf{x}_k$, we denote $\mathbf{x}_{k+1,u} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$ to be the point generated by the unconstrained BB method and $\mathbf{g}_{k+1,u}$ is the gradient at $\mathbf{x}_{k+1,u}$. Then the $\mathbf{x}_{k+1}$ generated by the PBB method is $\mathbf{x}_{k+1} = P(\mathbf{x}_{k+1,u})$. To simplify the calculation of $\mathbf{x}_{k+1,u}$, we decompose the Hessian as $A = Q^T D Q$, where

$$A = \begin{pmatrix} t+1 & t-1 \\ t-1 & t+1 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 \\ 0 & 2t \end{pmatrix}, \quad Q = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

and consider the orthogonal transformation $\bar{\mathbf{x}} = Q\mathbf{x}$. Correspondingly, we define $\bar{\mathbf{g}} = Q\mathbf{g}$. Since $\mathbf{g} = A\mathbf{x}$, it is easy to show that $\mathbf{g}_{k+1,u} = (I - \alpha_k A)\mathbf{g}_k$ and hence

$$\bar{\mathbf{g}}_{k+1,u} := Q\mathbf{g}_{k+1,u} = (I - \alpha_k D)\bar{\mathbf{g}}_k.$$

Noting that

$$\bar{\mathbf{g}}_1 = Q\mathbf{g}_1 = QA\mathbf{x}_1 = DQ\mathbf{x}_1 = -2\sqrt{2}\begin{pmatrix} 2 \\ t \end{pmatrix},$$

it follows from this and $\alpha_1 = 1/(t+1)$ that

$$\bar{\mathbf{g}}_{2,u} = (I - \alpha_1 D)\bar{\mathbf{g}}_1 = \frac{2\sqrt{2}(t-1)}{t+1}\begin{pmatrix} -2 \\ t \end{pmatrix}$$

and hence

$$\mathbf{x}_{2,u} = A^{-1}\mathbf{g}_{2,u} = A^{-1}Q^{-1}\bar{\mathbf{g}}_{2,u} = Q^T D^{-1}\bar{\mathbf{g}}_{2,u} = \frac{t-1}{t+1}\begin{pmatrix} -1 \\ 3 \end{pmatrix}.$$

It follows from $t \geq 31$ that $\mathbf{x}_2 = \mathbf{x}_{2,u}$, $\mathbf{g}_2 = \mathbf{g}_{2,u}$ and $\bar{\mathbf{g}}_2 = \bar{\mathbf{g}}_{2,u}$. Noting that

$$\alpha_2 = \frac{\mathbf{s}_1^T\mathbf{s}_1}{\mathbf{s}_1^T\mathbf{y}_1} = \frac{\bar{\mathbf{g}}_1^T\bar{\mathbf{g}}_1}{\bar{\mathbf{g}}_1^T D\bar{\mathbf{g}}_1} = \frac{t^2+4}{2(t^3+4)},$$

we can similarly obtain

$$\begin{aligned}
\mathbf{x}_3 = \mathbf{x}_{3,u} &= \frac{-2(t-1)^2}{(t+1)(t^3+4)}\begin{pmatrix} t^2+2 \\ -t^2+2 \end{pmatrix}, \\
\mathbf{x}_4 = \mathbf{x}_{4,u} &= \frac{2(t-1)^3}{(t+1)(t^3+4)^2}\begin{pmatrix} -t^4+8 \\ t^4+8 \end{pmatrix}, \\
\mathbf{x}_{5,u} &= \frac{-8t(t-1)^4}{(t+1)(t+4)(t^3+4)^2}\begin{pmatrix} t^2+2 \\ -t^2+2 \end{pmatrix}.
\end{aligned}$$

The point $\mathbf{x}_{5,u}$ is infeasible and we need to make a projection to obtain $\mathbf{x}_5$, giving

$$\mathbf{x}_5 = P(\mathbf{x}_{5,u}) = \begin{pmatrix} \dfrac{-8t(t^2+2)(t-1)^4}{(t+1)(t+4)(t^3+4)^2} \\ 1 \end{pmatrix}.$$

Finally, we can calculate $\mathbf{x}_{6,u}$ in the same way as for $\mathbf{x}_{4,u}$ and $\mathbf{x}_{5,u}$. The analytical expression of $\mathbf{x}_{6,u}$ is not written here since it is complicated and we only require that the projection $\mathbf{x}_6$ of $\mathbf{x}_{6,u}$ is exactly $\mathbf{x}_1$, which is readily verified numerically if t is suitably chosen. In the case that $\mathbf{x}_6 = \mathbf{x}_1$, we know that $\mathbf{s}_5$ is parallel to the horizontal axis and hence $\alpha_6 = 1/(t+1) = \alpha_1$. Therefore $\mathbf{x}_{5+i} = \mathbf{x}_i$ for $i \geq 1$, showing that the projected BB method will cycle between the five points.

Our numerical experiments with $t = 100$ show that $\mathbf{x}_6 = \mathbf{x}_1$ provided that the value $t \geq 31$. This example is illustrated in Figure 1, where $\mathbf{x}_{6,u} \approx (-4.2485, -3.2732)$ and the $\mathbf{x}_i$ $i = 1, \ldots, 5$ are approximately

$$\begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} -0.98020 \\ 2.9406 \end{pmatrix}, \begin{pmatrix} -1.9412 \\ 1.9404 \end{pmatrix}, \begin{pmatrix} -1.9214 \\ 1.9214 \end{pmatrix}, \begin{pmatrix} -0.073174 \\ 1 \end{pmatrix},$$

respectively.

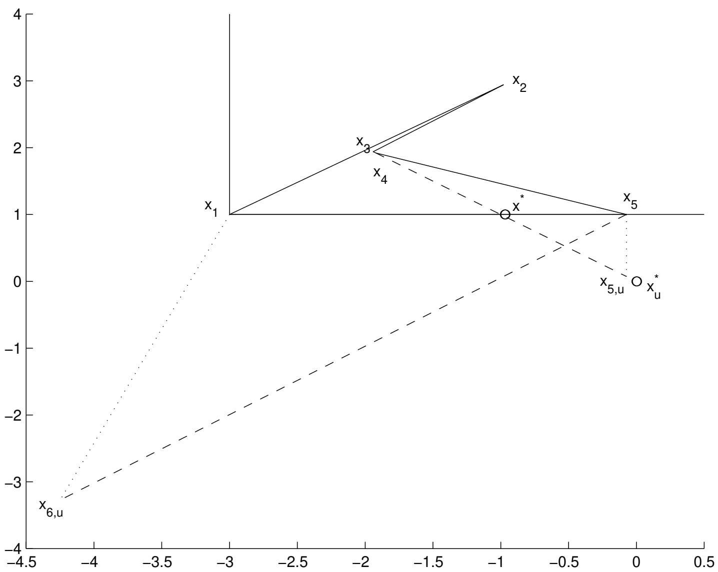

Fig. 1 A cyclic example of the PBB method

We have also tried different values for the initial steplength α_1 and find that the above example is not sensitive to the choice α_1. For example, in the case that $t = 100$, if $\alpha_1 = 0.005$ and $\alpha_1 = 0.02$ are used, then the projected BB method produces the same cycle from the sixth iteration. In addition, if a relatively large value is provided for α_1, then the method converges but takes many steps to get close to the optimal solution. Figure 2 plots the points $\{\mathbf{x}_8, \mathbf{x}_9, \ldots\}$ generated by the method for the example with $t = 100$ and $\alpha_1 = 0.175$.

It is worth noting that the above example is not suitable for the PABB method to cycle. The iterations $\{\mathbf{x}_i : i = 1, \ldots, 5\}$ generated by the PABB method are similar to those by the PBB method, namely

$$\begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} -0.98020 \\ 2.9406 \end{pmatrix}, \begin{pmatrix} -1.9412 \\ 1.9404 \end{pmatrix}, \begin{pmatrix} -1.9214 \\ 1.9214 \end{pmatrix}, \begin{pmatrix} -0.073160 \\ 1 \end{pmatrix},$$

respectively. However, the PABB method takes a smaller steplength $\alpha_5 = 5.4416\text{E-}3$ from (1.7), whereas $\alpha_5 = 4.5578\text{E-}2$ in the PBB method. The smaller steplength is such that $\mathbf{x}_6 \approx (-0.5717, 1)^T$ is a relative interior point of the face $\{(a, 1)^T : a \geq -3\}$ and hence the next step gives $\mathbf{x}_7 = (-\frac{99}{101}, 1)^T$, which is the optimal solution of the problem. However ther exists a counter-example showing that it is possible for the PABB method to cycle and not converge. This is the problem

$$\begin{array}{ll} \min & \frac{1}{200}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^T \begin{pmatrix} 3664 & -4752 \\ -4752 & 6436 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 60 \\ 80 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ s.t. & \qquad -40 \leq x_1 \leq 40, \quad x_2 \leq 300. \end{array}$$

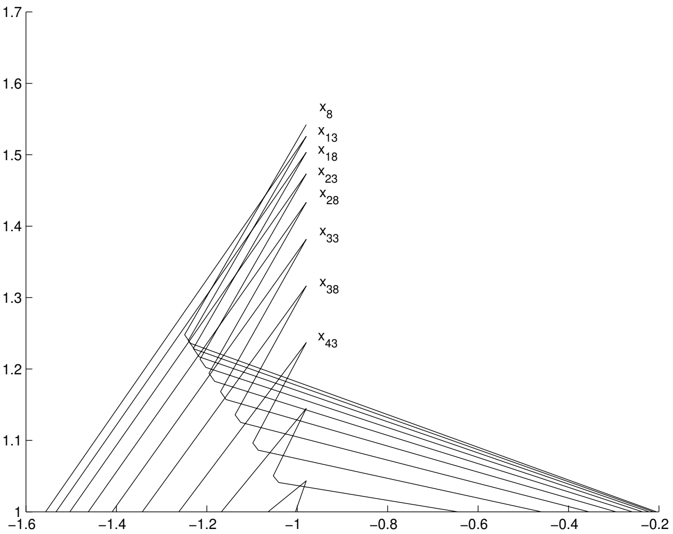

Fig. 2 A variation of the example of the PBB method

The PABB method cycles between the following eight points $\mathbf{x}_1$ to $\mathbf{x}_8$

$$\begin{pmatrix} -40 \\ -44.591 \end{pmatrix}, \begin{pmatrix} -40 \\ 300 \end{pmatrix}, \begin{pmatrix} 40 \\ 85.927 \end{pmatrix}, \begin{pmatrix} 40 \\ 45.663 \end{pmatrix},$$

$$\begin{pmatrix} 40 \\ 34.420 \end{pmatrix}, \begin{pmatrix} 40 \\ 28.291 \end{pmatrix}, \begin{pmatrix} 38.178 \\ 28.291 \end{pmatrix}, \begin{pmatrix} 35.054 \\ 25.927 \end{pmatrix},$$

with $\alpha_1 = 0.45261$, that is the steplength computed by (1.6) based on $\mathbf{x}_1$ and $\mathbf{x}_8$.

4. An adaptive nonmonotone line search

The counter-examples in the previous section show that the PBB and PABB methods without line searches may cycle between several points. Although our practical experience suggests that such behaviour is atypical, it is nonetheless prudent to modify the method by incorporating some sort of line search, so as to ensure global convergence in all cases. However it is important that the line search does not degrade the performance of the unmodified method.

Suppose that the current iteration is $\mathbf{x}_k$ and the step given by a projected gradient method is $\mathbf{d}_k = P(\mathbf{x}_k - \alpha_k \mathbf{g}_k) - \mathbf{x}_k$. Many methods have used an Armijo-type line search with acceptability test

$$f(\mathbf{x}_k + \lambda \mathbf{d}_k) \le f_r + \theta \lambda \mathbf{g}_k^T \mathbf{d}_k, \tag{4.1}$$

in which a decreasing sequence of values of $\lambda > 0$, is tried, starting with $\lambda = 1$, until the test is satisfied. Here f_r is some reference function value used for comparison, and $\theta \in (0,1)$ is a given constant. One advantage of a line search in the form (4.1) is that only one projection is required on each line search. It is easy to see that the unmodified PBB/PABB method corresponds to (4.1) with $f_r = +\infty$. In this case, there is no guarantee of global convergence due to the counter-examples in §3. If $f_r = f(\mathbf{x}_k)$, then the line search (4.1) causes the objective function values $f(\mathbf{x}_k)$ to decrease monotonically and global convergence can be proved. However, in this case the non-monotonic behaviour of BB-type methods is lost and the good numerical performance of the PBB/PABB method is seriously degraded.

Another possible choice for f_r is that suggested in the Grippo-Lampariello-Lucidi (GLL) nonmonotone line search [18]. For a given integer M, at the $k-$th iteration, this line search calculates f_r as follows

$$f_r = \max\{f(\mathbf{x}_{k-i}):\ 0 \le i \le \min\{k, M-1\}\}. \tag{4.2}$$

This GLL line search was first incorporated with the BB method for unconstrained optimization in [25] and good numerical results are reported on a wide range of test problems. Furthermore, [4] develops two different versions (SPG1 and SPG2) of the GLL line search for use with the projected BB method for the minimization of differentiable functions on closed convex sets. The value of $M = 10$ is recommended in both [25] and [4]. As will be shown below, this line search with a relatively small value of M can still degrade the unmodified BB method to a certain extent, although it does ensure global convergence.

To see by how much the performance of the PBB method is degraded for BQP problems, a BQP test problem derived from the large-scale quadratic problem of Fletcher [13] is used, which is based on the 3D Laplacian problem on a box. If a standard 7-point finite difference stencil is used, the matrix A can be defined by

$$A = \begin{bmatrix} W & -I & & & \\ -I & W & -I & & \\ & -I & W & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & W \end{bmatrix},$$

where

$$W = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & -I & T & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{bmatrix}, \quad T = \begin{bmatrix} 6 & -1 & & & \\ -1 & 6 & -1 & & \\ & -1 & 6 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 6 \end{bmatrix}.$$

Here T is $l \times l$, W is block $m \times m$ and A is block $n \times n$ where l, m, n are the number of interior nodes in each coordinate direction. The interval length in each direction is taken to be $h = 1/(l+1)$. Hence the dimensions of the box are $1 \times Y \times Z$ where $Y = (m+1)h$ and $Z = (n+1)h$. We fix the unconstrained solution $\mathbf{u}^*$ to the problem to be function

$$u(x,y,z) = x(x-1)y(y-Y)z(z-Z)\exp(-\frac{1}{2}\sigma^2(((x-\alpha)^2+(y-\beta)^2+(z-\gamma)^2)),$$

evaluated at the nodal points. The right hand side vector is then taken to be $\mathbf{b} = A\mathbf{u}^*$. Similarly to [13], we choose $l = m = n = 100$ giving a problem with 10^6 variables. The parameters σ, α, β, and γ are chosen in two different ways, that is

$$(a)\ \sigma = 20,\ \ \alpha = \beta = \gamma = 0.5 \qquad (b)\ \sigma = 50,\ \ \alpha = 0.4,\ \ \beta = 0.7,\ \ \gamma = 0.5.$$

To construct the upper and lower bounds of the BQP problem, we denote $u_{\max} = \|\mathbf{u}^*\|_\infty$ and impose the bound

$$-ru_{\max} \le x_i \le ru_{\max}, \quad i = 1,\ldots,N = lmn,$$

where $r > 0$ is some scalar. A range of values of r is considered, with small values of r giving rise to BQP problems with many active bounds, large values giving few active bounds, and $r = +\infty$ giving an unconstrained problem.

Table 4 lists the numerical results for different versions of the PBB method, where #it, #fc, #ls denote respectively the numbers of iterations, function evaluations and line searches required by each version. (By number of line searches we mean the number of iterations on which the unit step $\lambda = 1$ in (4.1) is not accepted.) The starting point is $\mathbf{u}_1 = \mathbf{0}$. A line search is always carried out on the first iteration of each version since the BB steplength is initially arbitrary (see §5 for further details). From Table 4 we see that for the line search used by the SPG2 method with $M = 10$, substantially more iterations and function evaluations are required, when compared with the unmodified PBB method. The performance of the SPG2 method may be improved somewhat by setting $M = 100$. However, this choice still degrades the unmodified PBB method to a noticeable extent.

Because of this behaviour we introduce another kind of nonmonotone line search. It uses an idea similar to one described by Toint [27] in a nonmonotone trust region method for convex programming, and subsequently applied by Dai and Zhang [11] to the BB method for unconstrained optimization. The numerical results reported in [11] show that this kind of line search is particularly suitable for the BB method in the nonquadratic case. The method again has a reference function value f_r, and each iteration must improve on the reference value. The method involves a small integer parameter $L > 0$, and f_r is reduced if the method fails to improve on the previous best value of f in at most L iterations. We dispense with the requirement (such as in (4.1), for example) to obtain a *sufficient reduction* in f, since in real computation any reduction is bounded uniformly away from zero by a small amount, and this is sufficient to ensure global convergence. We refer to this kind of line search as an *adaptive nonmonotone line search*.

For box constrained quadratic programming, the following simplified adaptive nonmonotone line search is suitable for the PBB method. Let us denote by f_{best}

Problem	r	Unmodified		SPG2($M=10$)			SPG2($M=100$)			Adaptive($L=10$)		
		#it	#fc	#it	#fc	#ls	#it	#fc	#ls	#it	#fc	#ls
Laplace (a)	0.1	499	503	664	974	106	749	779	15	499	503	0
	0.2	815	820	827	1282	140	1047	1118	30	1020	1026	1
	0.4	1306	1311	1571	2490	304	1445	1536	40	1306	1311	0
	0.6	1408	1414	1439	2262	268	1812	1928	59	1082	1097	3
	1.2	692	698	979	1558	176	857	938	27	786	803	4
	5	461	468	867	1374	161	815	927	43	720	733	5
	20	497	503	1015	1677	196	487	554	21	537	555	6
	∞	816	822	1244	2109	239	1137	1239	36	816	822	0
Laplace (b)	0.1	1205	1208	1240	1930	231	1284	1348	38	1166	1177	7
	0.2	1123	1127	1089	1681	191	1071	1119	18	1002	1008	2
	0.4	1106	1111	1301	2080	251	1003	1063	26	1106	1111	0
	0.6	1219	1224	1585	2436	285	1716	1823	49	1286	1310	14
	1.2	679	684	1221	1980	222	896	996	35	679	684	0
	5	561	567	1060	1746	203	782	854	29	839	847	2
	20	855	861	1068	1798	201	745	824	28	530	538	1
	∞	577	583	1117	1813	218	731	809	30	577	583	0

Table 4. Testing different line searches for the PBB method

the current least value of the objective function over all past iterates, that is, at the k-th iteration

$$f_{\text{best}} = \min_{1\le i\le k} f(\mathbf{x}_i).$$

The number of iterations since the value of f_{best} was obtained is denoted by l. Also we define the candidate function value f_c to be the maximum value of the objective function since the value of f_{best} was found.

Now let us see how to determine the reference function f_r. Suppose that L is a preset positive integer. Initially, we can set $f_r = +\infty$. This choice of f_r allows $f(\mathbf{x}_k) \ge f(\mathbf{x}_1)$ on early iterations. (As observed by Fletcher [13], requiring $f(\mathbf{x}_k) < f(\mathbf{x}_1)$ on early iterations may degrade the performance of the unmodified BB method in the unconstrained case.) If the method can find a better function value in L iterations, then the value of f_r remains unchanged. Otherwise, if $l = L$, we reset the reference function value f_r to f_c and reset f_c to the current value $f(\mathbf{x}_k)$. We initialize $f_r = +\infty$, $f_{\text{best}} = f_c = f(\mathbf{x}_1)$, and $l = 0$. The strategy to update f_r may be described in matlab-style notation as follows.

```
if f_k < f_best,
   f_best=f_k, f_c=f_k, l=0,
else
   f_c=max{f_c,f_k}, l=l+1,
   if l=L, f_r=f_c, f_c=f_k, l=0, end
end
```

Problem	r	Unmodified		SPG2($M=10$)			SPG2($M=100$)		
		#it	#fc	#it	#fc	#ls	#it	#fc	#ls
Laplace (a)	0.1	356	360	363	396	22	356	368	3
	0.2	549	554	445	492	25	635	656	5
	0.4	524	529	529	581	26	632	650	5
	0.6	578	584	712	783	37	613	634	6
	1.2	610	616	645	719	34	665	684	4
	5	649	656	520	574	27	689	710	6
	20	551	557	725	800	38	591	615	7
	∞	736	742	561	614	30	563	580	2
Laplace (b)	0.1	616	619	637	674	27	755	771	6
	0.2	749	753	614	653	25	563	579	6
	0.4	493	498	569	622	29	523	542	6
	0.6	720	725	646	706	37	664	680	4
	1.2	766	771	635	725	42	839	859	6
	5	870	876	597	661	34	597	621	8
	20	601	607	586	671	37	762	780	4
	∞	658	664	768	823	32	730	751	7

Table 5. Testing different line searches for the PABB method

A simple argument proves global convergence in real arithmetic. It can be seen that if f_{best} is updated an infinite number of times then global convergence occurs. Assume the contrary that f_{best} is unchanged for all k sufficiently large. In this case there exists an infinite subsequence of iterations k_i on which $l = L$ and f_r is reset to f_c. Now $f_c < f_r$ because f_c is a recent value of f_k for which $f_k < f_r$. Thus the values of f_r that are reset on iterations k_i are strictly monotonically decreasing. Hence there exists a subsequence on which f_k decreases without bound, which contradicts the fact that f_{best} is unchanged. We note that the above algorithm differs a little from that proposed by Toint [27], in that $f_c = f$ is reset when $l = L$, which is necessary for establishing our convergence result, whereas Toint obtains global convergence in a somewhat different way.

Numerical results are also reported in Table 4 for the PBB method using the adaptive nonmonotone line search with $L = 10$. We can see that very few line searches are needed, and the numerical results are closely similar to the unmodified PBB method.

Different versions of the PABB method have also been tested and the numerical results are listed in Table 5. The numerical results using the adaptive nonmonotone line search with $L = 10$ are not tabulated explicitly, since they are exactly the same as the unmodified version. Thus the aim of finding a modification which requires no line searches at all (excluding the first iteration) has been achieved in this case. In fact we could have used the smaller value of $L = 4$ and the same conclusion would hold. In other words, for these problems the unmodified PABB method is always able to find a smaller function value in at most 4 iterations. Overall, the unmodified (or $L = 10$ version) wins in 10 of the 16 cases, as against the better of the GLL nonmonotone line search with either $M = 10$ or $M = 100$, although the discrepancy is not quite as large as in Table 4. When compared with the results in Table 4, we see that the PABB method performs better than the PBB method and is not so much influenced by the line search.

To sum up, we feel that the adaptive nonmonotone line search preserves the properties of the unmodified PBB and PABB methods very well. At the same time, we also see that in several cases, the number of iterations and function evaluations can be usefully decreased if a few line searches are done. One example is the PBB method with the adaptive line search with $L = 10$ for Laplace(a) with $r = 0.6$. Therefore it still remains to study whether there exists some line search strategy which can not only ensure the global convergence but also improve the performance of the unmodified PBB and PABB methods uniformly.

5. Further Numerical Experiments

In this section, we further test the unmodified PBB and PABB methods against the GLL nonmonotone line search, the adaptive nonmonotone line search, and the GPCG conjugate gradient method. Some practical problems in [22] and random test problems are used in these tests.

We use the same linesearch procedure as in [4]. The constant θ in (4.1) is set to 10^{-4}. The initial value of λ in (4.1) is 1. If the current λ does not satisfy the line search condition, then we do a quadratic interpolation to obtain a new trial value λ_{new}. In the case that $\lambda \leq 0.1$ or $\lambda_{new} \notin [0.1, 0.9\lambda]$, we simply set $\lambda_{new} = 0.5\lambda$. The following stopping condition, recommended in [22], is used in all our tests:

$$\|\nabla q_\Omega(\mathbf{x}_k)\|_2 \leq 10^{-5}\|\mathbf{g}_1\|_2,$$

where $\nabla q_\Omega(\mathbf{x}_k)$ is given in (2.4). Since the BB steplength is not defined on the first iteration, in our algorithm we set the initial trial steplength to be $\alpha_1 = \|\nabla q_\Omega(\mathbf{x}_k)\|_\infty^{-1}$. Similarly to the algorithm [4], we set $\alpha_k = \alpha_{\max}$ if $\mathbf{s}_{k-1}^T\mathbf{y}_{k-1} \leq 0$ and restrict $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ for other α_k's. The values $\alpha_{\min}$ and $\alpha_{\max}$ are 10^{-30} and 10^{30}, respectively.

Firstly, we test different versions of the PBB and PABB methods for some practical problems from [22] for which A is positive definite. We number the obstacle problem (A), obstacle problem (B), and elastic-plastic problem as Problems 1, 2, 3 (see Table 6). For Problem 3, as in [22] we use three values of c namely 5, 10, and 20. Results from a variety of different initial points are given in the table, where $\mathbf{e}$ denotes the n-dimensional vector whose components are all one, $\mathbf{0}$ is the origin, $\mathbf{l}$ and $\mathbf{u}$ are the corresponding lower and upper bounds varying with the problem, and $\mathbf{v} = \frac{1}{2}(\mathbf{l} + \mathbf{u})$.

Table 6 shows the number of function evaluations for the PBB and PABB methods using the two different nonmonotone line searches. For both the PBB and PABB

P	n	$nfree$	$\mathbf{x}_1$	PBB		PABB		GPCG
				$M=10$	$L=10$	$M=10$	$L=10$	
1	5625	2122	**l**	211(14)	187	154(6)	131	119
1	5625	2122	**e**	251(20)	310	202(8)	147	146
1	10000	3843	**l**	345(24)	296	200(7)	245	157
1	10000	3843	**e**	583(41)	537	272(8)	206	211
1	15625	6108	**l**	466(36)	410	324(8)	180	188
1	15625	6108	**e**	868(66)	595	310(14)	346	238
2	5625	4171	**l**	257(15)	171	126(6)	148	131
2	5625	4171	**u**	207(11)	231	174(8)	165	120
2	5625	4171	**v**	247(15)	218	148(7)	136	90
2	10000	7588	**l**	365(26)	329	195(6)	237	174
2	10000	7588	**u**	432(26)	359	195(8)	205	168
2	10000	7588	**v**	211(13)	199	172(6)	193	127
2	15625	11990	**l**	598(40)	433	358(8)	275	234
2	15625	11990	**u**	601(40)	367	237(10)	320	216
2	15625	11990	**v**	509(37)	350	324(13)	213	177
3(5)	5625	3953	**0**	862(63)	451	373(12)	273	244
3(5)	5625	3953	**u**	511(32)	357	355(8)	377	221
3(5)	10000	7016	**0**	1091(78)	897	550(19)	433	307
3(5)	10000	7016	**u**	1077(67)	915	553(18)	417	326
3(5)	15625	10961	**0**	$\geq$2000(132)	1406	668(20)	628	402
3(5)	15625	10961	**u**	1637(113)	1049	511(10)	557	385
3(10)	5625	2073	**0**	163(13)	246	136(3)	202	140
3(10)	5625	2073	**u**	210(12)	167	126(3)	144	111
3(10)	10000	3632	**0**	359(25)	377	218(6)	229	174
3(10)	10000	3632	**u**	290(17)	289	217(5)	197	139
3(10)	15625	5789	**0**	314(22)	480	261(9)	288	217
3(10)	15625	5789	**u**	605(44)	372	265(10)	234	167
3(20)	5625	1025	**0**	76(4)	126	66(1)	88	67
3(20)	5625	1025	**u**	77(2)	75	70(1)	66	50
3(20)	10000	1768	**0**	115(6)	145	89(3)	133	82
3(20)	10000	1768	**u**	118(5)	92	105(2)	80	59
3(20)	15625	9248	**0**	97(5)	241	82(2)	133	113
3(20)	15625	9248	**u**	159(8)	105	130(3)	120	73

Table 6 Numerical results for some practical problems in [22]

methods, use of the adaptive nonmonotone line search with $L = 10$ requires no line searches after the first iteration, thus repeating the outcome of the unmodified methods. In fact, for the PABB method, the same conclusion holds for any $L \geq 4$. Thus the adaptive nonmonotone line search continues to give results that are close to the unmodified method. For the PBB and PABB methods using the GLL nonmonotone line seach with $M = 10$, the number of line searches is listed in brackets after the number of function evaluations. In the $M = 10$ case the PBB method is significantly worse. The most efficient algorithm overall (excluding the GPCG method) is the PABB method using the adaptive nonmonotone line search with $L = 10$, although in the 3(10) and 3(20) cases, there is not much to choose between the $M = 10$ and $L = 10$ line searches.

For the above practical problems, [22] provides numerical results for the GPCG method which combines the gradient projection technique and the conjugate gradient method. In Table 6 we show the number of Hessian-vector products needed by the GPCG method. Exactly how to measure these results against those for the PABB method is not entirely clear. On a one-to-one basis however, the PABB method with $L = 10$ is inferior, but not by an amount more than about 50%. Thus the PABB method is not totally uncompetitive, and there is some reason to think that for box constrained non-quadratic problems, the PABB method may prove to be a promising choice.

Secondly, we have tested the methods on some random SPD test problems generated in a way similar to [21]. We follow [21] in using the five parameters n, $ncond$, $ndeg$, $na(\mathbf{x}^*)$, and $na(x_1)$. In particular, we denote $A = PDP^T$ where

$$P = (I - 2\mathbf{w}_3\mathbf{w}_3^T)(I - 2\mathbf{w}_2\mathbf{w}_2^T)(I - 2\mathbf{w}_1\mathbf{w}_1^T),$$

and where $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ are random unit vectors, and D is a diagonal matrix whose i-th component is defined by

$$\log d_i = \frac{i-1}{n-1} ncond, \quad i = 1, \ldots, n.$$

The parameter $ncond$ in the above relation specifies the condition number of A. In our tests, we have fixed $n = 10^4$ and $ncond = 6$.

We generate a solution $\mathbf{x}^*$ with components chosen randomly in the interval $(-1, 1)$. The choice of active set $\mathcal{A}(\mathbf{x}^*)$ depends on the integer parameter $na(\mathbf{x}^*)$. Random numbers μ_i in $(0, 1)$ are generated for $i = 1, \ldots, n$ and i is selected for $\mathcal{A}(\mathbf{x}^*)$ if $\mu_i \leq \frac{na(\mathbf{x}^*)}{n}$. This algorithm is also used to select the active constraints $\mathcal{A}(\mathbf{x}_1)$ at the starting point on the basis of the parameter $na(\mathbf{x}_1)$. Components of $\mathbf{x}_1$ that are not in $\mathcal{A}(\mathbf{x}_1)$ are set to $\frac{l_i + u_i}{2}$.

The values of $\mathbf{l}$, $\mathbf{u}$ and $\mathbf{b}$ can now be determined using the parameter $ndeg$. This parameter specifies the extent to which the resulting problem will be close to being dual degenerate, in the way described in [21]. We determine the value of $\nabla q(\mathbf{x}^*) = \mathbf{r}$

by setting $|r_i| = 10^{-\mu_i ndeg}$ for $i \in \mathcal{A}(\mathbf{x}^*)$, where μ_i is randomly generated in $(0, 1)$. Finally we set $\mathbf{b} = A\mathbf{x}^* - \mathbf{r}$ and define

$$l_i = -1, \quad u_i = +1, \quad r_i = 0,$$

for $i \notin \mathcal{A}(\mathbf{x}^*)$, and

$$l_i = x_i^*, \quad u_i = +1, \quad r_i > 0,$$

or

$$l_i = -1, \quad u_i = x_i^*, \quad r_i < 0,$$

for $i \in \mathcal{A}(\mathbf{x}^*)$.

Table 7 shows the number of function evaluations required by the GPCG method and by several projected gradient methods. Listed in brackets in the columns headed PBB($M = 10$) and PABB($M = 10$) are the number of iterations. For the GPCG method, we also provide the number of Hessian-vector products in brackets similarly to [21]. From Table 7, similar conclusions can be drawn as from Table 6. The GPCG method requires fewer function evaluations and Hessian-vector products, but the projected gradient methods are not totally uncompetitive and can solve the problems in a reasonable number of iterations. As in Table 6, for both the PBB and PABB methods with the adaptive nonmonotone line search, we again find that no line searches are done from the second iteration. Likewise this property again holds for the PABB method provided that $L \geq 4$. The PABB method is again more efficient than the PBB method. In addition, we also see from Table 7 that all these methods are not much influenced by the parameter $ndeg$.

Finally, we have tested the methods for random indefinite problems. In this case, a parameter $negeig$ is used to specify the number of negative eigenvalues of A. Given an integer $negeig \in [1, n]$, we generate random numbers μ_i in $(0, 1)$ for $i = 1, \ldots, n$ and change the sign of the i-th element of D if $\mu_i \leq negeig/n$. If the diagonal matrix D has l entries less than zero, then the active set $\mathcal{A}(x^*)$ of any local minimizer x^* has at least l entries. In addition, we still use the parameters n, $ncond$, and $na(\mathbf{x}_1)$ to generate the matrix A, the vector $\mathbf{b}$, and the starting point $\mathbf{x}_1$ as before. The feasible set is defined by setting $\mathbf{l} = -\mathbf{e}$ and $\mathbf{u} = \mathbf{e}$. Again, we fix $n = 10^4$ and $ncond = 6$ but change the parameters $na(\mathbf{x}_1)$ and $negeig$.

In the indefinite case, some modifications are necessary for the PABB method and the GPCG method. For the PABB method, we use the formula (1.6) at the $k-$th iteration if $\mathbf{s}_{k-2}^T\mathbf{y}_{k-2} \leq 0$ happens and hence $\alpha_{k-1} = \alpha_{\max}$. We change to using (1.7) if both $\mathbf{s}_{k-1}^T\mathbf{y}_{k-1} > 0$ and the previous steplength α_{k-1} is calculated by (1.6). For the GPCG method, it is possible that $\mathbf{p}_k^T A\mathbf{p}_k \leq 0$ and hence the steplength either in the gradient projection procedure or in the conjugate gradient procedure is not well defined. In this case, we remedy the method by setting $\alpha_k = \|\nabla q_\Omega(\mathbf{x}_k)\|_\infty^{-1}$. Further, if the same happens in the conjugate gradient procedure, we then terminate the procedure and do a line search on the next iteration. The numerical results in Table 8 show that the modified GPCG method works well.

$na(\mathbf{x}^*)$	$na(\mathbf{x}_1)$	$ndeg$	GPCG	PBB($M=10$)	$L=10$	PABB ($M=10$)	$L=10$
1000	1000	1	43(105)	173(152)	109	131(126)	122
		3	40(107)	176(153)	145	129(114)	124
		6	30(100)	171(154)	127	130(121)	116
		9	45(101)	170(152)	149	121(112)	127
1000	5000	1	35(107)	186(165)	155	132(123)	145
		3	36(105)	157(136)	143	135(129)	110
		6	40(98)	180(157)	169	116(112)	117
		9	34(106)	178(155)	157	135(124)	124
1000	9000	1	42(100)	183(166)	147	123(113)	122
		3	45(116)	160(137)	124	155(150)	118
		6	44(106)	169(147)	131	119(115)	137
		9	42(106)	198(177)	135	142(134)	125
5000	1000	1	32(100)	129(111)	134	150(139)	114
		3	46(110)	159(140)	124	127(123)	130
		6	76(118)	209(188)	147	134(125)	117
		9	78(111)	205(186)	138	136(123)	135
5000	5000	1	41(107)	173(155)	131	126(114)	104
		3	43(116)	194(165)	130	145(138)	107
		6	81(124)	211(189)	167	160(145)	143
		9	83(118)	168(148)	158	146(137)	129
5000	9000	1	32(110)	154(137)	117	128(125)	111
		3	53(109)	184(166)	162	134(130)	104
		6	63(107)	188(161)	168	151(140)	137
		9	91(130)	174(153)	168	167(155)	216
9000	1000	1	30(92)	149(133)	111	149(138)	90
		3	41(91)	140(120)	103	121(118)	114
		6	84(120)	203(186)	145	114(108)	116
		9	91(129)	134(118)	151	148(137)	107
9000	5000	1	27(92)	168(149)	144	112(105)	114
		3	66(106)	164(145)	137	136(132)	97
		6	90(127)	148(128)	130	120(113)	130
		9	89(124)	154(136)	167	133(128)	111
9000	9000	1	40(107)	200(171)	139	123(112)	114
		3	53(111)	185(165)	126	131(122)	124
		6	80(116)	204(182)	132	144(138)	115
		9	88(126)	143(124)	156	121(112)	127

Table 7. Numerical results for random SPD test problems

$na(\mathbf{x}_1)$	$negeig$	GPCG	PBB($M=10$)	$L=10$	PABB ($M=10$)	$L=10$
1000	1000	268(208)	196(170)	158	253 (193)	180
	2500	235(203)	179(157)	165	234 (211)	229
	5000	277(202)	199(176)	151	225 (195)	142
	9000	232(176)	205(190)	240	283 (252)	183
5000	1000	254(174)	221(185)	165	248 (217)	351
	2500	219(209)	161(136)	256	160 (143)	157
	5000	375(331)	174(149)	174	177 (163)	181
	9000	338(230)	288(257)	178	153 (140)	156
9000	1000	244(174)	215(180)	174	168 (147)	473
	2500	253(194)	177(153)	179	185 (156)	169
	5000	243(195)	313(262)	233	365 (279)	242
	9000	371(236)	264(219)	215	184 (155)	118

Table 8. Numerical results for random indefinite test problems

methods	1	2	3	4	5
winners	17	6	50	7	40
1 : i	—	26:94	20:100	23:97	21:99
i : 3	20:100	15:105	—	28:92	54:66

Table 9. Winners of the methods according to function value

Although Table 8 suggests that the PBB method using the adaptive nonmonotone line search with $L = 10$ provides relatively good performance, it is difficult to read too much into this comparison since different methods may find different local minimizers. In addition, it still remains to study what the best adaption of the GPCG method in [22] is to the indefinite case. Therefore we only conclude that all of these methods can provide a solution in a reasonable number of iterations. In addition, we note that fewer line searches are done in both the PBB and PABB methods with the adaptive nonmonotone line search with $L = 10$.

In our numerical experiments, however, we observe quite often that the PBB and PABB method reach a point at which function value is less that the point generated by the GPCG method. We have run each pair $(na(\mathbf{x}_0), negeig)$ 10 times and so have obtained 120 groups of function values. In Table 9 we show comparisons between various methods numbered 1-5, which stand for GPCG, PBB with $M = 10$, PBB with $L = 10$, PABB with $M = 10$, and PABB with $L = 10$, respectively. Given a collection of methods $\mathcal{S} \subset \{1, 2, 3, 4, 5\}$, we say that the method i is the winner for some problem if it provides the least function value amongst all the methods in $\mathcal{S}$. From the second line of Table 9, we can see that the PBB method with $L = 10$ is the best. Further, if we compare the winners of the GPCG method with other methods separately (see the third line of the table), the GPCG method wins only about 20 of the problems. The separate comparison of the PBB method with $L = 10$ with other methods (see the fourth line) further shows that this method has a strong ability to get a better solution.

6. Concluding remarks

In this paper, we have carefully considered using new gradient projection methods for BQP problems, both in cases where the matrix A is positive definite, and where it may be indefinite. Since the projected BB-like methods can be used in both the identification of the optimal face and the exploration of the face, a procedure to judge where the optimal face is reached is not necessary. Consequently, the new gradient projection methods are easy to code and work well in practice.

Although both the PBB and PABB methods without line searches are usually successful for solving BQP problems and perform very well, there is no theory to guarantee global convergence. Two-dimensional counter-examples have been constructed such that the methods cycle between several points. Therefore some kind of line search is shown to be necessary to ensure the global convergence of these methods.

In the case that the GLL line search is applied, the performance of any of the unmodified methods is noticeably degraded, although this effect is not so severe if a large value of M is used (but this may be unsatisfactory if the methods are used to solve non-quadratic box constrained problems). This has led us to consider the use of a kind of adaptive nonmonotone line search. The technique of choosing the reference function value adaptively is similar to that proposed in [27] and applied to the BB method for unconstrained optimization in [11]. The basic idea of the line search is to keep the reference function value f_r unchanged if a smaller function value is found in L iterations, and otherwise to reset f_r to the maximum function value since the last overall best function was found. A difference between the adaptive line search of this paper and the adaptive techniques in [27] and [11] is that here the initial reference function value f_r is set to be $+\infty$. Our numerical experiments show that our new adaptive line search can preserve the good performance of the unmodified PBB/PABB method very well, particularly for the PABB method. Thus for the PBB method using the adaptive line search with $L = 10$, only a few line searches are required, whereas the choice $L = 4$ is usually sufficient for the PABB method.

For positive definite BQP problems, the PABB method is seen to perform better than the PBB method. However, in the indefinite case, the PBB method is preferred since the differences between the two methods are not great but the PBB method more often finds a solution with smaller function value. Due to the success of the adaptive line search and the alternate BB formula (1.8), it is likely that that projected gradient algorithms will prove to be very effective for minimizing non-quadratic functions subject to box constraints or general convex constraints. In a forthcoming paper, we shall extend this work to design effcient projected gradient methods with adaptive line search for solving the large-scale quadratic programming problems subject to box constraints and one single linear constraint, as this is another situation in which the projection operation can be efficiently carried out.

References

[1] H. Akaike, *On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method*, Ann. Inst. Statist. Math. Tokyo, 11 (1959), pp. 1-17.

[2] J. Barzilai and J. M. Borwein, *Two-point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141-148.

[3] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Transactions on Automatic Control AC-21 (1976), pp. 174-184.

[4] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. Optim. 10 (2000), pp. 1196-1211.

[5] E. G. Birgin, J. M. Martínez, and M. Raydan, *Inexact Spectral Projected Gradient methods on convex sets*, 2003 (to appear in: IMA Journal on Numerical Analysis).

[6] M. Brendel, *On two algorithms for the minimization of box constrained quadratic functions*, MS344 honours project, Department of Mathematics, University of Dundee, 2001.

[7] Y. H. Dai, *Alternate step gradient method*, Research report, 2001.

[8] Y. H. Dai and R. Fletcher, *On the asymptotic behaviour of some new gradient methods*, Research report NA/212, Department of Mathematics, University of Dundee, 2003.

[9] Y. H. Dai and L.-Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, IMA J. Numer. Anal., 26 (2002), pp. 1-10.

[10] Y. H. Dai, J. Y. Yuan, Y. Y. Yuan, *Modified Two-point stepsize gradient methods for unconstrained optimization*, Computational Optimization and Applications 22 (2002), pp. 103-109.

[11] Y. H. Dai and H. C. Zhang, *An adaptive two-Point stepsize gradient algorithm*, Numerical Algorithms, 27 (2001) 377-385.

[12] R. S. Dembo and U. Tulowitzki, *On the minimization of quadratic functions subject to box constraints*, Working Paper 71, School of Organization and Management, Yale University, New Haven, CT, 1983.

[13] R. Fletcher, *On the Barzilai-Borwein Method*, Research report, Department of Mathematics, University of Dundee, 2001.

[14] A. Friedlander and J. M. Martínez, *On the maximization of a concave quadratic function with box constraints*, SIAM J. Optimization 4 (1994), pp. 204-212.

[15] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, *Gradient method with retards and generalizations*, SIAM J. Numer. Anal., 36 (1999), 275-289.

[16] E. Galligani, V. Ruggiero, L. Zanni, *Variable projection methods for large-scale quadratic optimization in data analysis applications*, in: Equilibrium Problems and Variational Models, F. Giannessi, A. Maugeri and P. M. Pardalos, eds., Nonconvex Optim. and Its Applic., Kluwer Academic PUbl., to appear. (2002)

[17] A. A. Goldstein, *Convex programming in Hilbert space*, Bulletin of the American Mathematical Society 70 (1964), pp. 709-710.

[18] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707-716.

[19] L. Grippo and M. Sciandrone, *Nonmonotone globalization techniques for the Barzilai-Borwein gradient method*, Computational Optimization and Applications 23 (2002), pp. 143-169.

[20] E. S. Levitin and B. T. Polyak, *Constrained minimization problems*, USSR Computational Mathematics and Mathematical Physics 6 (1966), pp, 1-50.

[21] J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377-400.

[22] J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93-113.

[23] B. T. Polyak, *The conjugate gradient method in extremal problems*, USSR Comput. Math. and Math. Phys., 9 (1969), pp. 94-112.

[24] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal., 13 (1993), pp. 321-326.

[25] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7 (1997), pp. 26-33.

[26] T. Serafini, G. Zanghirati, L. Zanni, *Gradient Projection Methods for Quadratic Programs and Applications in Training Support Vector Machines*, Research report, 2003.

[27] Ph. L. Toint, *A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints*, Math. Prog. 77 (1997), pp. 69-94.

[28] S. J. Wright, *Implementing proximal point methods for linear programming*, Journal of Optimization Theory and Applications, 65 (1990), pp. 531-554.

[29] E. K. Yang and J. W. Tolle, *A class of methods for solving large, convex quadratic programs subject to box constraints*, Math. Prog., 51 (1991), pp. 223-228.

[30] G. Zanghirati, L. Zanni, *A parallel solver for large quadratic programs in training support vector machines*, Parallel Computing, to appear. (2003).