Dynamic choice of the leaving–face criterion in bound–constrained quadratic minimization *

Maria A. Diniz–Ehrhardt [†] Márcia A. Gomes–Ruggiero [‡]
Sandra A. Santos [§]

Abstract

In this work we focus our attention on the quadratic subproblem of trust–region algorithms for bound–constrained minimization, proposing and testing dynamic choices for the parameter in charge of the decision of leaving or not the current face of the feasible set. The practical consequences of an appropriate decision of such parameter have shown to be crucial, particularly when dual degenerate and ill-conditioned problems are solved.

Key words and phrases: Numerical tests, Large–scale problems, Bound–constrained quadratic minimization

[*] This work was supported by FAPESP (Grants 90–3724–6 and 95–6498–8), CNPq, FINEP and FAEP–UNICAMP.

[†] Department of Applied Mathematics, IMECC – UNICAMP, University of Campinas, CP 6065, 13081–970 Campinas SP, Brazil (cheti@ime.unicamp.br).

[‡] Department of Applied Mathematics, IMECC – UNICAMP, University of Campinas, CP 6065, 13081–970 Campinas SP, Brazil (marcia@ime.unicamp.br).

[§] Department of Mathematics, IMECC – UNICAMP, University of Campinas, CP 6065, 13081–970 Campinas SP, Brazil (sandra@ime.unicamp.br).

1 Introduction

In a previous work [8] we compared the numerical performance of the software `BOX-QUACAN` [13] with the package `LANCELOT` [7] for the solution of large–scale bound–constrained minimization problems:

$$\begin{array}{cc} \text{Minimize} & f(x) \\ \text{s.t.} & \underline{\ell} \leq x \leq \underline{u}\,, \end{array} \tag{1}$$

where $f : I\!R^n \to I\!R$ is differentiable on the feasible set and any component of the bounds $\underline{\ell}, \underline{u}$ may be infinite. Both algorithms are based on the trust–region approach: at the j-th iteration, a quadratic model for the decrease of the objective function is built around the current point x_j:

$$f(x_j + s) - f(x_j) \approx q_j(s) \equiv \frac{1}{2} s^T B_j s + g_j^T s \tag{2}$$

where $g_j \equiv \nabla f(x_j)$, $B_j \in I\!R^{n \times n}$ and $B_j = B_j^T$.

Since the quadratic model (2) becomes less representative as the step s increases in size, we can trust in approximating $f(x_j + s) - f(x_j)$ by $q_j(s)$ in a neighborhood of x_j, that is, in the set

$$\Omega = \{ s \in I\!R^n \mid \underline{\ell} \leq x_j + s \leq \underline{u},\ \|s\| \leq \Delta \}$$

where $\Delta > 0$ and $\|\cdot\|$ is an arbitrary norm in $I\!R^n$. Thus, an approximate minimizer $\widehat{s}$ of $q_j(s)$ in the region Ω is a good candidate for step. In other words, $x_j + \widehat{s}$ is accepted and defined as x_{j+1} as long as there is a sufficient decrease from $f(x_j)$ to $f(x_j + \widehat{s})$. Otherwise, the step $\widehat{s}$ is rejected, the size of set Ω is decreased by reducing the trust–region radius Δ and a new quadratic subproblem is defined. In the comparison made in [8] the norm $\|\cdot\|$ was the ℓ_∞ norm so that the step $\widehat{s}$ remains bound constrained. `BOX-QUACAN` and `LANCELOT` turned out to be competitive, with a superior performance of the former for quadratic problems.

`BOX-QUACAN` requires Fortran routines for computing the objective function value, its gradient, its Hessian times a vector and a driver for setting the data and the parameters. `LANCELOT` demands the problem to be coded in `SIF` (Standard Input Format), so that its interface generates the necessary Fortran routines. If the user is already familiar with coding in `SIF`, both `BOX-QUACAN` and `LANCELOT` can be used. `BOX-QUACAN` and its interface for decoding `SIF` are available under request to the authors.

The quadratic solver `QUACAN` [3, 13] was developed for dealing with the whole feasible set and approximately solving the quadratic subproblem by combining conjugate gradients with projected gradients and a mild active set strategy. We observed that in a few quadratic problems, the number of iterations of `QUACAN` was rather large, and we concluded that this behavior was due to a specific parameter $\eta \in (0, 1)$, set by the user, which defines the leaving–face criterion. If η is close to 1, the current face is fully exploited and if η is close to 0 the current face can be abandoned prematurely. Through some tests with the fixed choices $\eta \equiv 0.1$ and $\eta \equiv 0.9$ we became aware of the potential improvement of adopting a dynamic choice for the leaving face criterion.

In this work we propose two heuristics for deciding to stay in the current face or to leave it. In both of them the decision is based on the values of the Euclidean norm of the projected gradient and its orthogonal components: the *chopped* and the *internal* gradient. The chopped and internal gradients give measures of progress of the quadratic function outside and inside the current face, respectively.

This paper is organized as follows: in Section 2 we summarize the comparative numerical performance of BOX-QUACAN and LANCELOT [8]. In Section 3 we review the main ingredients for bound-constrained quadratic minimization, including the statement of the algorithm implemented in the solver QUACAN. Two heuristics for the leaving-face criterion are described in Section 4. Numerical results are presented and analyzed in Section 5, in which the two heuristics are compared with the performances of keeping η fixed at the values 0.1 and 0.9 and also of solving the problems using LANCELOT. Finally, in Section 6 some conclusions and ideas for future research are presented.

2 Comparative numerical performance of BOX-QUACAN and LANCELOT

In [8] the numerical performance of the algorithm BOX-QUACAN is compared with the package LANCELOT, developed by Conn, Gould and Toint [7]. BOX-QUACAN was put in a context by the solution of a set of 220 problems of minimizing a general nonlinear function subject to simple bounds. These problems were extracted from the CUTE collection [4], so that specific features of both approaches could be compared and analyzed into the same environment. Due to their trust–region nature, both algorithms have many similarities, but no doubt the philosophy behind the quadratic solver is the main difference between them. In fact, the two of them only require matrix–vector products for dealing with the box constrained subproblems, but BOX-QUACAN was developed for exploiting the subproblems to a great extent, dealing with the whole feasible set by combining conjugate gradients (or another iterative solver [12]) with projected gradients and an active set strategy specially designed so that many constraints can be added or dropped in a single iteration. In LANCELOT, on the other hand, conjugate gradients are applied just in a convenient portion of the feasible set, defined by the generalized Cauchy point [7].

Figures 1 and 2 summarize the average computational effort of algorithms BOX-QUACAN and LANCELOT. It is shown the geometric means of the number of iterations performed by each quadratic solver (inner iterations) and of the gradient evaluations, which represent the distinct points generated by each algorithm (outer iterations). Figure 1 stresses the value of using a specially designed solver for simple bounded quadratic problems that exploits the whole feasible set, instead of relying on the identification information provided by the generalized Cauchy point. With a convenient setting of parameters, for quadratic problems, BOX-QUACAN performs a single outer iteration. Figure 2 points out that BOX-QUACAN and LANCELOT are competitive as far as non-quadratic problems are concerned.

An interesting aspect of the quadratic solver QUACAN was detected through the real bound-constrained quadratic problem ODNAMUR, with dimension 11130. To uniformize the choices for the whole set of tests, we defined very loosely the parameter in charge of deciding to abandon the current face, setting $\eta \equiv 0.1$, as suggested by the authors [3]. For problem ODNAMUR, however, such choice showed to be rather poor. Probably due to

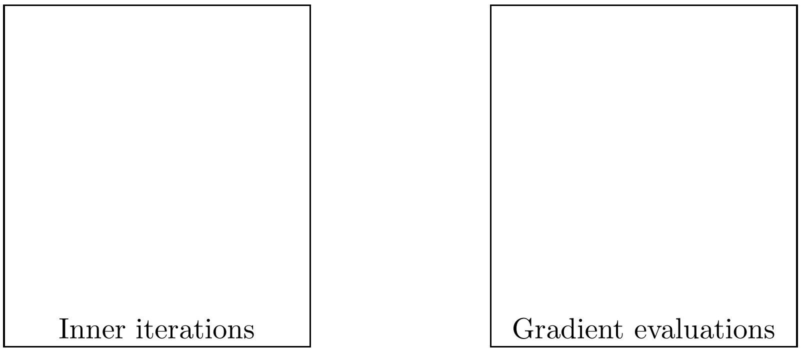

Figure 1: Computational effort of quadratic problems

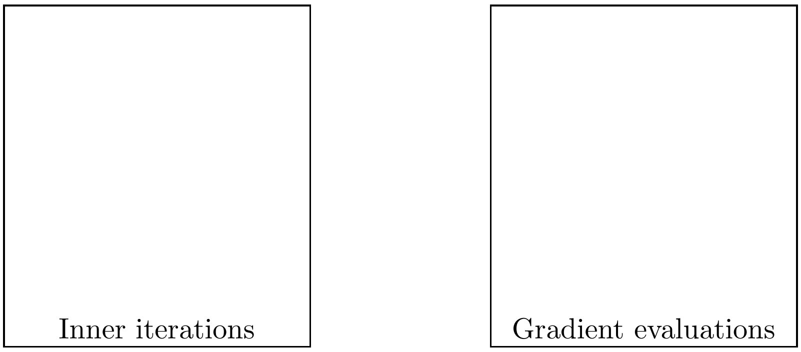

Figure 2: Computational effort of non-quadratic problems

dual degeneracy, for this problem the best policy was to investigate better the current face before abandoning it, to avoid wastes in premature leaving and having to go back to it. With $\eta \equiv 0.9$ the number of `QUACAN` iterations was reduced by 68%, a very expressive improvement, as shown in Table 1.

Software	Inner iterations	Gradient evaluations
`BOX-QUACAN` ($\eta = 0.1$)	117902	1
`BOX-QUACAN` ($\eta = 0.9$)	37846	1
`LANCELOT`	51556	10

Table 1: The bound–constrained quadratic problem `ODNAMUR`

It is worth mentioning that `BOX-QUACAN` performed better than `LANCELOT` for the quadratic problems (total of 53, 17 of which unconstrained and 36 simple–bounded) with the loose choice $\eta \equiv 0.1$, being problem `ODNAMUR` the only exception. Anyway, the degenerated feature of this problem motivated us to investigate dynamic strategies for selecting η, since the behavior of `QUACAN` has shown to be sensitive to the choice of this parameter.

3 Bound–constrained quadratic minimization

From now on we focus our attention on the problem

$$\begin{array}{ll} \text{Minimize} & q(s) \equiv \frac{1}{2} s^T B s + g^T s \\ \text{s.t.} & \ell \leq s \leq u \,, \end{array} \tag{3}$$

where $B = B^T \in I\!R^{n \times n}$, $g, \ell, u \in I\!R^n$ and in general n is large. There are efficient techniques for solving (3) based on gradient projections [2] and conjugate gradients [15], in which the feasible box is split into disjoint faces, the conjugate gradient method is applied within the faces (where the problem is essentially unconstrained) and the polygonal path obtained by the projections of the half-lines defined by suitable descent directions is used for leaving the current face whenever necessary (see [11, 16]). The convergence results for methods of this type are as follows: in [16] convergence is proved in the case of a strictly convex quadratic and finite convergence is proved when the limit point is not dual degenerate. The properties of the *chopped gradient*, introduced in [10], allowed the authors to prove in [11] finite convergence in the convex case even for a singular Hessian and in the presence of dual degeneracy. In [12], convergence is proved for (not necessarily strictly) convex bound-constrained quadratic minimization, using the Barzilai-Borwein method [1, 17] within the faces. In [3], where general quadratics are considered, the bound of the norm of the quadratic Hessian, essential ingredient in the finite convergence of degenerate problems, present in [11], is replaced by another condition which ensures global convergence even in the presence of dual degeneracy and finite identification in the nondegenerate case.

For completeness, in the following some definitions and notations will be introduced. Denoting by $\Omega = \{s \in I\!R^n \mid \ell \leq s \leq u\}$ the feasible set of problem (3), an *open face of* Ω

is a set $F_I \subset \Omega$ such that I is a (possibly empty) subset of $\{1, 2, \ldots, 2n\}$ such that i and $n+i$ cannot belong simultaneously to I for any $i \in \{1, 2, \ldots, n\}$ and such that

$$F_I = \{s \in \Omega \mid s_i = \ell_i \text{ if } i \in I, s_i = u_i \text{ if } n+i \in I, \ell_i < s_i < u_i \text{ otherwise}\}.$$

The closure of each open face will be denoted by $\overline{F}_I$, $[F_I]$ is the smallest linear manifold that contains F_I, $S(F_I)$ is the parallel subspace to $[F_I]$ and dim F_I is the dimension of $S(F_I)$. Clearly, dim $F_I = n - |I|$, where $|I|$ denotes the number of elements of the set I.

For each $s \in \Omega$, the *(negative) projected gradient* $\overline{g}_P(s) \in I\!R^n$ is defined componentwise by

$$\overline{g}_P(s)_i = \begin{cases} 0 & \text{if} \quad s_i = \ell_i \quad \text{and} \quad \frac{\partial q}{\partial s_i}(s) > 0 \\ 0 & \text{if} \quad s_i = u_i \quad \text{and} \quad \frac{\partial q}{\partial s_i}(s) < 0 \\ -\frac{\partial q}{\partial s_i}(s) & \text{otherwise}\,. \end{cases}$$

A necessary condition for s being a global solution of (3) (sufficient in the convex case) is that $\overline{g}_P(s) = 0$.

For each $s \in \overline{F}_I$, the *internal gradient* $\overline{g}_I(s) \in I\!R^n$ and the *chopped gradient* $\overline{g}_I^C(s) \in I\!R^n$ are defined componentwise, respectively, by

$$\overline{g}_I(s)_i = \begin{cases} 0 & \text{if} \quad i \in I \quad \text{or} \quad n+i \in I \\ -\frac{\partial q}{\partial s_i}(s) & \text{otherwise} \end{cases}$$

and

$$\overline{g}_I^C(s)_i = \begin{cases} 0 & \text{if} \quad i \notin I \quad \text{or} \quad n+i \notin I \\ 0 & \text{if} \quad i \in I \quad \text{and} \quad \frac{\partial q}{\partial s_i}(s) > 0 \\ 0 & \text{if} \quad n+i \in I \quad \text{and} \quad \frac{\partial q}{\partial s_i}(s) < 0 \\ -\frac{\partial q}{\partial s_i}(s) & \text{otherwise}\,. \end{cases}$$

It is worthwhile noticing that $\overline{g}_I(s)$ is the orthogonal projection of $-\nabla q(s)$ on $S(F_I)$ and $\overline{g}_P(s) = \overline{g}_I(s) + \overline{g}_I^C(s)$.

The method for solving bound-constrained quadratic minimization which is considered in this paper, implemented in the subroutine `QUACAN`, is described in [3], together with its convergence properties. In [9] a similar approach independently developed is presented. Previous work with related ideas can be found in [10, 11, 12, 13, 16].

The algorithm implemented in the subroutine `QUACAN` produces a sequence $\{s_k\}$ of approximations to the solution of (3) based on a partial minimization of the quadratic in the different visited faces. As s_k belongs to a face F_I, an "internal algorithm" for minimizing unconstrained quadratics is activated, working with the variables that are free in F_I. The main assumption on this internal algorithm is to be convergent in the sense of either finding, in a finite number of steps, a point outside Ω (but, naturally, in $[F_I]$) or that any limit point is stationary to

$$\begin{array}{ll} \text{Minimize} & q(s) \\ \text{s.t.} & s \in [F_I]\,. \end{array} \tag{4}$$

In other words, such algorithm either finds a stationary point which belongs to F_I or violates the bounds that are inactive in face F_I. In order to verify at each step of the internal algorithm how close the generated sequence is of a stationary point, the norms of the chopped and of the projected gradients are compared. If the ratio between them $\left(\frac{\|\overline{g}_I^C(s)\|}{\|\overline{g}_P(s)\|}\right)$ is large (the maximum value is 1), it means that the internal gradient is small compared with the chopped gradient, and so it seems of little value to remain in the current face, being worth leaving the active set F_I. This change of face is made by means of the chopped gradient direction. A sequence generated in this fashion has proved to be convergent to a stationary point of (3) ([3]), solution of the problem in the convex case.

Since the internal algorithm plays an essential role in the performance of the method used for solving (3), we state its main features. An algorithm for (4)(essentially an unconstrained problem) has *good properties for bound-constrained minimization* whenever it produces a sequence (maybe finite) $z_0, z_1, z_2, \ldots \in [F_I], z_0 \in F_I$, satisfying:

(a) If z_ν and $z_{\nu+1}$ are defined then $q(z_{\nu+1}) < q(z_\nu)$.

(b) If $z_{\nu+1}$ is not defined (i.e. the sequence ends in z_ν) then either z_ν is a stationary point of (4) or a direction d has been found such that $\lim_{\lambda\to\infty} q(z_\nu + \lambda d) = -\infty$. In this case, if $z_\nu + \lambda d \in \Omega$ for all λ, then (3) does not have a solution. Otherwise, if $z_\nu + \lambda d \notin \Omega$ for large λ it is chosen a breakpoint $\overline{\lambda}$ and set $z_{\nu+1} = z_\nu + \overline{\lambda} d \in \Omega$ such that $q(z_{\nu+1}) < q(z_\nu)$, and the sequence generated by the internal algorithm stops in $z_{\nu+1}$.

(c) If $\{z_\nu\}$ is an infinite sequence, then every limit point is stationary of (4). If it does not have limit points ($\|z_\nu\| \to \infty$) then $\lim_{\nu\to\infty} q(z_\nu) = -\infty$ holds.

Since the conjugate gradient method for minimizing unconstrained quadratics either achieves a stationary point in a finite number of steps or generates a direction along of which the quadratic goes to minus infinity, it satisfies (a), (b) or (c) above. In other words, conjugate gradient has good properties for bound-constrained minimization and is actually used in the implementation of subroutine `QUACAN`. In [3] other iterative methods are studied that satisfy these properties under certain circumstances.

In the following the algorithm implemented in the subroutine `QUACAN` is stated. The notation $P[x, S]$ for the orthogonal projection of x on the set S is used. The computational way of detecting that problem (3) does not have a solution is by means of a safeguard in the number of iterations.

3.1 The algorithm for bound constrained quadratic minimization

Let $\eta \in (0, 1)$ be given independently of k, let $s_0 \in \Omega$ be an arbitrary initial point and let $I = I(s_0)$ be such that $s_0 \in F_I$. Starting with $k = 0$ and $\nu = 0$, the steps of the algorithm are:

Step 1. *(Stopping criterion)*

If $\|\overline{g}_P(s_k)\| = 0$, stop.

```
Step 2. (Test if current face must be left or not)
    If
        $\|\overline{g}_I^C(s_k)\| > \eta\|\overline{g}_P(s_k)\|$        (5)
    then $d_\nu = \overline{g}_I^C(s_k)$ and go to Step 4.

Step 3. (Direction $d_\nu$ is obtained by one CG-iteration applied to (4))
            If $\nu = 0$ then
                $d_\nu = \overline{g}_I(s_k)$
            else
                obtain $\beta$ from CG-algorithm
                $d_\nu = \overline{g}_I(s_k) + \beta d_{\nu-1}$.

Step 4. (Search along the polygonal path $P[s_k + \lambda d_\nu, \Omega]$, $\lambda \geq 0$)
    Obtain $\overline{\lambda}$ so that $q(P[s_k + \overline{\lambda} d_\nu, \Omega]) < q(s_k)$.

Step 5. (Prepare to the next iteration)
            $s_{k+1} = P[s_k + \overline{\lambda} d_\nu, \Omega]$
            If $I(s_{k+1}) \neq I(s_k)$ then
                $I = I(s_{k+1})$
                $\nu = 0$
            else
                $\nu = \nu + 1$.

Step 6. (Updates)
    Set $k = k + 1$, update $\overline{g}_P(s_k)$, $\overline{g}_I(s_k)$, $\overline{g}_I^C(s_k)$ and go to Step 1.
```

4 Heuristics for Leaving the Current Face

We observe that the test (5) of Step 2 of Algorithm 3.1, which decides between leaving or not the current active set, strongly determines the behavior of the algorithm. In order to improve the performance of the subroutine `QUACAN`, we consider convenient choices of the parameter η, allowing its modification during the process of (approximately) solving (3). We propose two heuristics for leaving the current face. Both of them are based on the norms of the projected gradient and its orthogonal components: the chopped and the internal gradients.

4.1 Heuristic H_α

Our first idea is to perform the test (5) of Algorithm 3.1 defining η according to

$$\eta = 1 - \left(\frac{\|\overline{g}_P(s_k)\|}{\|g_{P_T}\|} \right)^{\alpha} \ , \ \eta \in [0.1, 0.9] \ , \tag{6}$$

where α is a positive real number and g_{P_T} is a typical value for $\overline{g}_P$ (e.g. for a chosen integer ν, $g_{P_T} \equiv \overline{g}_P(s_\nu)$ if $k > \nu$ and $\eta \equiv 0.1$ if $k \leq \nu$).

Figure 3: Heuristic H_α

Figure 3 shows the curves (6) in terms of $\|\overline{g}_P(s_k)\|$, with $\alpha = 0.5, \alpha = 1.0$ and $\alpha = 2.0$. Combining condition (5) with the formula (6) for η, we can see that using $\alpha = 0.5$ the tendency to abandon the face is more frequent than using $\alpha = 2.0$.

4.2 Piecewise linear heuristic

Another idea to decide between staying in the actual face or leaving it is based on the evaluation of the following piecewise linear function:

$$p(t) = \begin{cases} 0.90t & \text{if} \quad 0 \le t \le a \\ 0.75t + 0.15a & \text{if} \quad a \le t \le b \\ 0.50t + 0.25b + 0.15a & \text{if} \quad b \le t \le c \\ 0.10t + 0.40c + 0.25b + 0.15a & \text{if} \quad t \ge c \ , \end{cases} \tag{7}$$

where $a = K_1 \|g_{P_T}\|, b = \|g_{P_T}\|$ and $c = K_2 \|g_{P_T}\|$. Taking $t = \|\overline{g}_P(s_k)\|$, the algorithm leaves the face when $\|\overline{g}_I^C(s_k)\| > p(t)$ and stays in it otherwise. The typical value g_{P_T} can be modified during the process at each ξ iterations, $\xi > 0$, whenever $\|\overline{g}_P(s_k)\| < \|g_{P_T}\|$.

Figure 4: Piecewise linear heuristic

Figure 4 shows the piecewise linear function $p(t)$ and the straight lines associated with condition (5) for $\eta = 0.9\,(r_1)$, $0.7\,(r_2)$, $0.5\,(r_3)$ and $0.1\,(r_4)$. In the figure it is specified the region for which the algorithm leaves the current face or stays in it (above and below the piecewise linear function $p(t)$, respectively). It is worth noticing that the value of η which characterizes a tie in the decision of leaving the current face or staying in it is $1/\sqrt{2}$ and not $1/2$. In fact, since Euclidean norms are being used, if $\|\overline{g}_I^C(s_k)\| = \|\overline{g}_I(s_k)\|$ then $\|\overline{g}_P(s_k)\| = \sqrt{2}\|\overline{g}_I^C(s_k)\|$.

If we compare this piecewise linear heuristic with condition (5) for $\eta \equiv 0.1$, we can observe that the former has a greater tendency to stay in the current face when $\|\overline{g}_P(s_k)\|$ is close to zero. On the other hand, comparing the heuristic with condition (5) using $\eta \equiv 0.9$, we note that, if $\|\overline{g}_P(s_k)\|$ is still sufficiently large, the algorithm is going to leave the face more frequently when the piecewise linear heuristic is used.

5 Numerical Results

To analyze the performance of the subroutine QUACAN with the two heuristics proposed in Section 4, we ran a set of 26 bound–constrained quadratic problems from the CUTE collection. All of them have more than 1000 variables, since we are interested in large–scale problems. They are distributed in three sets, according to their origin: academic problems (that are proposed by researchers to test their algorithms); real problems (that result of practical applications); and modelling problems (that are part of modelling exercises).

The tests were developed in Fortran 77 double precision with the -O compiler option and run in a SUN Ultra 1 Creator. We used an interface [14] to run BOX-QUACAN with CUTE.

In Tables 3–5 we report the number of iterations performed by QUACAN with the different criteria to leave the current face of the feasible set. The first column has the name of the problem, according to CUTE, and its dimension N. The second column shows the total number of inner iterations performed by LANCELOT. The next columns show the number of QUACAN iterations when we used: condition (5) with $\eta \equiv 0.1$ and $\eta \equiv 0.9$; the heuristic H_α with $\nu = 5$ and $\alpha = 1.0$ $(H_\alpha 1), \alpha = 0.5$ $(H_\alpha 2)$ and $\alpha = 2.0$ $(H_\alpha 3)$; and the piecewise linear heuristic, represented by $PLH1$,$PLH2$,$PLH3$ and $PLH4$. In this latter heuristic we used $g_{P_T} = \overline{g}_P(s_0)$ and the choices defined in Table 2.

	K_1	K_2	ξ
$PLH1$	0.1	10	10
$PLH2$	0.1	10	50
$PLH3$	0.1	10	∞
$PLH4$	0.01	10	10

Table 2: Parameter choices for the piecewise linear heuristics

With these different parameters we observe that in heuristics $PLH2$ and $PLH3$, we kept $g_{P_T} \equiv \overline{g}_P(s_0)$ during many iterations; since $\|\overline{g}_P(s_0)\|$ is, in general, larger than $\|\overline{g}_P(s_k)\|$, for all $k > 0$, these heuristics give a tendency to the algorithm to remain in some faces. On the other hand, with the heuristic $PLH4$, the tendency of the algorithm is to abandon the current face more frequently. The heuristic $PLH1$ has an intermediate behavior.

For all the problems, the sequence $\{s_k\}$ generated by QUACAN converged to a stationary point (we considered s_k a stationary point if $\|\overline{g}_P(s_k)\| < 10^{-5}$), and for each problem the same solution was obtained for each different heuristic, except Chenhark. For this problem, using condition (5) with $\eta \equiv 0.1$, the subroutine QUACAN achieved a local minimizer different from the one obtained with the other choices for η (the same obtained by LANCELOT.

PROBLEM (N)	LANCELOT	$\eta \equiv 0.1$	$\eta \equiv 0.9$	$H_\alpha 1$	$H_\alpha 2$	$H_\alpha 3$	$PLH1$	$PLH2$	$PLH3$	$PLH4$
BIGGSB1 (1000)	66509	3545	3610	3713	3745	3713	3978	3499	3609	4342
BQPGAUSS (2003)	9511	7442	6363	6507	6252	6345	6038	6225	6350	6096
CHENHARK (1000)	136276	3730	17	20	20	20	17	17	17	16
JNLBRNG1 (15625)	2556	1142	973	1354	1438	1188	1491	1262	1303	1408
JNLBRNG2 (15625)	2673	1106	935	888	837	917	872	904	935	771
JNLBRNGA (15625)	2135	1179	483	631	732	613	453	454	485	436
JNLBRNGB (15625)	4439	2661	3554	3724	3506	3752	2755	2749	2933	2742
NOBNDTOR (14884)	1539	884	431	403	448	420	499	405	418	387
OBSTCLAE (15625)	7608	759	386	602	648	602	590	398	386	614
OBSTCLAL (15625)	805	305	251	252	298	273	261	254	247	270
OBSTCLBL (15625)	3259	366	375	322	378	352	410	375	404	381
OBSTCLBM (15625)	1483	242	199	204	202	225	212	181	198	205
OBSTCLBU (15625)	1102	443	313	307	341	319	338	300	281	345

Table 3: Academic bound–constrained quadratic problems

PROBLEM (N)	LANCELOT	$\eta \equiv 0.1$	$\eta \equiv 0.9$	$H_\alpha 1$	$H_\alpha 2$	$H_\alpha 3$	$PLH1$	$PLH2$	$PLH3$	$PLH4$
TORSION1 (14884)	1347	803	363	386	377	367	380	355	335	399
TORSION2 (14884)	5053	765	435	565	654	603	495	510	571	415
TORSION3 (14884)	390	270	164	170	192	184	179	175	187	177
TORSION4 (14884)	5954	225	165	170	183	200	186	185	181	168
TORSION5 (14884)	114	84	76	83	71	82	78	76	76	73
TORSION6 (14884)	7355	78	79	77	76	83	78	79	79	78
TORSIONA (14884)	1339	858	349	386	401	393	397	351	402	355
TORSIONB (14884)	5000	588	443	573	548	436	528	465	474	479
TORSIONC (14884)	390	273	183	181	176	164	184	196	193	168
TORSIOND (14884)	9430	227	178	183	180	168	175	194	183	176
TORSIONE (14884)	114	82	79	75	75	78	73	72	72	73
TORSIONF (14884)	5343	90	79	73	77	75	78	73	73	73

Table 4: Modelling bound–constrained quadratic problems

PROBLEM (N)	LANCELOT	$\eta \equiv 0.1$	$\eta \equiv 0.9$	$H_\alpha 1$	$H_\alpha 2$	$H_\alpha 3$	$PLH1$	$PLH2$	$PLH3$	$PLH4$
ODNAMUR (11130)	51556	117902	37846	37846	37370	37846	38359	40530	37846	37964

Table 5: Real bound–constrained quadratic problems

Table 6 summarizes the geometric means of the number of QUACAN iterations. This average was chosen to accomodate the very different and problem depending order of magnitude of the results. The notation used for the different criteria is similar to the previous tables. The numbers show that, except for LANCELOT and $\eta \equiv 0.1$, which demand larger effort, all the other choices have practically similar performances.

LANCELOT	$\eta \equiv 0.1$	$\eta \equiv 0.9$	$H_\alpha 1$	$H_\alpha 2$	$H_\alpha 3$	$PLH1$	$PLH2$	$PLH3$	$PLH4$
3014.088	660.604	386.261	412.926	425.807	415.268	409.709	392.873	394.526	392.260

Table 6: Geometric means of the number of QUACAN iterations

With the aim of analyzing the behavior of QUACAN with the different criteria for leaving the current face, we had registered when the algorithm QUACAN performed the smallest and the largest number of iterations with a certain criterion. Each column of Table 7 shows the number of problems in which the smallest (second row) and the largest (third row) number of QUACAN iterations were achieved with the criterion indicated at the first row of the table. The last row represents the difference between second and third rows, that is, the balance between the best and the worst performance of the algorithm QUACAN, with each criterion, in terms of its number of iterations. In this table, the results of LANCELOT and QUACAN with condition (5) and $\eta \equiv 0.1$ were not included because, in general, they were the worst.

CRITERION	$\eta \equiv 0.9$	$H_\alpha 1$	$H_\alpha 2$	$H_\alpha 3$	$PLH1$	$PLH2$	$PLH3$	$PLH4$
THE “BEST”	5	2	3	3	1	4	6	7
THE “WORST”	3	4	6	5	3	3	2	3
BALANCE	+2	−2	−3	−2	−2	+1	+4	+4

Table 7: Performance of QUACAN for each heuristic

In Figure 5, for each problem, numbered according to their order of appearance in Tables 3, 4 and 5, we plot the ratio between the number of QUACAN iterations of the best heuristic and of the one that keeps $\eta \equiv 0.9$. For 20 out of the 26 solved problems (77%) the ratio is smaller than one, mostly concentrated between 0.8 and 1.0. This shows the value of investigating other heuristics instead of simply using $\eta \equiv 0.9$.

To illustrate part of the results of Table 7, in Figure 6 we plot the ratio between the number of QUACAN iterations of $PLH4$ and $\eta \equiv 0.9$. For 14 out of the 26 solved problems (54%) the ratio is smaller than one. For problems BIGGSB1, JNLBRBG1 and OBSTCLAE, however, this ratio is around 1.2,1.4 and 1.6, respectively.

Problem ODNAMUR, which motivated this work, attained its best results with heuristic $H_\alpha 2$, although not expressively better than using $\eta \equiv 0.9$. In Figure 7 the comparative

Figure 5: Ratios between the number of QUACAN iterations of the best heuristic (smallest value) and of $\eta \equiv 0.9$.

results among all heuristics can be visualized, by plotting the ratio between the number of iterations of the quadratic solver with the strategy indicated and the one that keeps $\eta \equiv 0.9$.

6 Final Remarks

Tables 3-7 and Figures 5-7 confirmed the results shown in the bar charts of Section 2, that is, for any chosen criterion for leaving the face of the current feasible set, BOX-QUACAN performs better than LANCELOT in the solution of bound-constrained quadratic problems. This reinforces the value of exploiting the whole feasible set instead of resting upon the face identified by the generalized Cauchy point. Another evident conclusion among the proposed heuristics is that setting $\eta \equiv 0.1$, as suggested by the authors ([3]), is not a recommended policy for quadratic problems. In fact, according to Tables 3-5, 21 out of 26 problems (i.e. 81%) had the worst performance with $\eta \equiv 0.1$ in terms of QUACAN iterations compared with the other choices for η. Moreover, although by Table 6 the conservative choice $\eta \equiv 0.9$ produces best average results, the analysis of the behavior of the heuristics for each problem, as in Table 7, shows that it is worth investing in the family of piecewise linear ones.

A natural next step for this work is to investigate the performance of the algorithm BOX-QUACAN for minimizing a general nonlinear function with simple bounded variables

Figure 6: Ratios between the number of `QUACAN` iterations of the heuristic $PLH4$ and $\eta \equiv 0.9$.

when dynamic choices for the parameter η are allowed.

Acknowledgements: We are indebted to J.M. Martínez for helpful suggestions and to A.R. Conn, N.I.M Gould and Ph.L. Toint for making the package `LANCELOT` available for academic research.

References

[1] J. Barzilai & J. M. Borwein, Two point step size gradient methods, *IMA Journal of Numerical Analysis* **8**, pp. 141–148, 1988.

[2] D. P. Bertsekas, Projected Newton methods for optimization problems with simple constraints, *SIAM Journal on Control and Optimization* **20**, pp. 141–148, 1982.

[3] R. H. Bielschowsky, A. Friedlander, F. A. M. Gomes, J. M. Martínez & M. Raydan, An adaptive algorithm for bound constrained quadratic minimization, Technical Report, Institute of Mathematics, State University of Campinas, 1995 (to appear in *Investigación Operativa*).

[4] I. Bongartz, A. R. Conn, N. I. M. Gould & Ph. L. Toint, CUTE: Constrained and Unconstrained Testing Environment, *ACM Transactions on Mathematical Software* **21**, pp. 123–160, 1995.

Figure 7: Problem `ODNAMUR`: Ratios between the number of iterations of `LANCELOT` and `QUACAN` with the indicated heuristics and `QUACAN` with $\eta \equiv 0.9$.

[5] A.R. Conn, N.I.M. Gould & Ph.L. Toint, Global convergence of a class of trust region algorithms for optimization with simple bounds, *SIAM Journal on Numerical Analysis*, **25**, pp. 433–460, 1988. See also same journal **26**, pp. 764–767, 1989.

[6] A.R. Conn, N.I.M. Gould & Ph.L. Toint, Testing a class of methods for solving minimization problems with simple bounds on the variables, *Mathematics of Computation*, **50**, pp. 399–430, 1988.

[7] A.R. Conn, N.I.M. Gould & Ph.L. Toint, *LANCELOT - A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Verlag, Berlin, Heidelberg, New York, 1992.

[8] M. A. Diniz–Ehrhardt, M. A. Gomes–Ruggiero and S. A. Santos, Comparing the numerical performance of two trust-region algorithms for large-scale bound-constrained minimization, Technical Report **RP03-97**, Institute of Mathematics, State University of Campinas, 1997 (to appear in *Investigación Operativa*).

[9] Z. Dostál, Box constrained quadratic programmming with proportioning and projections, *SIAM Journal on Optimization* **7**, pp. 871–887, 1997.

[10] A. Friedlander & J. M. Martínez, On the numerical solution of bound constrained optimization problems, *RAIRO Operations Research* **23**, pp. 319–341, 1989.

[11] A. Friedlander & J. M. Martínez, On the maximization of a concave quadratic function with box constraints, *SIAM Journal on Optimization* **4**, pp. 177–192, 1994.

[12] A. Friedlander, J. M. Martínez & M. Raydan, A new method for large scale box constrained quadratic minimization problems, *Optimization Methods and Software* **5**, pp.57–74, 1995.

[13] A. Friedlander, J. M. Martínez & S. A. Santos, A new trust region algorithm for bound constrained minimization, *Applied Mathematics & Optimization* **30**, pp. 235–266, 1994.

[14] M. A. Gomes–Ruggiero & P. G. Kakuzo, Análise do conjunto de problemas CUTE em testes de software em otimização, Technical Report **RP07-96**, Institute of Mathematics, State University of Campinas, 1996.

[15] M. R. Hestenes & E. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards B* **49**, pp. 409–436, 1952.

[16] J. J. Moré & G. Toraldo, On the solution of large quadratic programming problems with bound constraints, *SIAM Journal on Optimization* **1**, pp. 99–113, 1991.

[17] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, *IMA Journal of Numerical Analysis* **13**, pp. 321–326, 1993.