Comparing the Numerical Performance of Two Trust–region Algorithms for Large–scale Bound–constrained Minimization*

Maria A. Diniz–Ehrhardt [1] *Márcia A. Gomes–Ruggiero* [1]
Sandra A. Santos [2]

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

[2]Department of Mathematics, IMECC-UNICAMP, University of Campinas
CP 6065, 13081-970 Campinas SP, Brazil
sandra@ime.unicamp.br

Abstract
In this work we compare the numerical performance of the software `BOX-QUACAN` *with the package* `LANCELOT`*. We put* `BOX-QUACAN` *in a context by means of solving an extensive set of problems, so that specific features of both approaches are compared. Through the computational results, conclusions are made about classes of problems for which each algorithm suits better and ideas for future research are devised.*

Keywords: Numerical tests, Trust–region methods, Large–scale problems, Bound–constrained minimization.

---

1 Introduction

In this work we consider the problem

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

where $f : I\!R^n \to I\!R$ is differentiable on the feasible set $\mathcal{B} = \{x \in I\!R^n \mid l \leq x \leq u\}$ and any component of the bounds l, u may be infinite. We focus our interest on

*This work was supported by FAPESP (Grant 90–3724–6), CNPq, FINEP and FAEP–UNICAMP.
Received december 1996, revised version accepted june 1997.

the case where n is large, since large–scale bound constrained problems appear frequently in applications. Furthermore, in the last few years, Conn, Gould and Toint [3, 4, 5, 6] stressed the importance of developing efficient algorithms for problem (1). They showed that general large–scale nonlinear programming problems can be efficiently solved using augmented Lagrangian techniques, as long as a good method for solving (1) is available.

In [11], Friedlander, Martínez and Santos proposed the algorithm `BOX-QUACAN`, of trust–region type for solving (1). At each iteration, `BOX-QUACAN` considers the subproblem of minimizing a (not necessarily convex) quadratic on a box which is the intersection of the feasible set $\mathcal{B}$ with a trust region defined by the ℓ_∞ norm.

Our aim is to compare the numerical performance of the algorithm `BOX-QUACAN` with the package `LANCELOT`, developed by Conn, Gould and Toint [6]. We intend to put `BOX-QUACAN` in a context by means of solving an extensive set of problems of type (1) from the `CUTE` collection [1], so that specific features of both approaches can be compared and analyzed. 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 `BOX-QUACAN` and `LANCELOT`. In fact, in both approaches only matrix–vector products are required 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 [10]) 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.

This work is organized as follows: in Section 2 the common features of trust–region algorithms applied to problem (1) are described. Sections 3 and 4 state, respectively, the distinctive characteristics of `BOX-QUACAN` and `LANCELOT`. The numerical results are presented and analysed in Section 5. Finally, in Section 6 some conclusions are stated and ideas for future research are devised.

2 Trust-region Algorithms for Bound-constrained Minimization

Feasible point methods are well suited to problem (1) due to the simplicity of the set $\mathcal{B}$. This is the case of trust–region algorithms, which can be either applied directly to (1) (cf. [3, 4, 11]) or combined with an interior–point approach, so that strictly feasible iterates are generated (cf. [2, 7]).

Roughly speaking, the trust–region method for solving (1) consists of the following. At the k-th iteration, a quadratic model for the decrease of the objective function is built around the current point x_k:

$$f(x_k + s) - f(x_k) \approx q_k(s) \equiv \frac{1}{2}s^T B_k s + g_k^T s \tag{2}$$

where $g_k \equiv \nabla f(x_k)$, $B_k \in I\!R^{n\times n}$ and $B_k = B_k^T$.

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

$$\mathcal{B}_\Delta = \{s \in I\!R^n \mid x_k + s \in \mathcal{B}, \|s\| \leq \Delta\}$$

where $\Delta > 0$ and $\|\cdot\|$ is an arbitrary norm in $I\!R^n$. Thus, an approximate minimizer $\widehat{s}$ of $q_k(s)$ in the region $\mathcal{B}_\Delta$ is a good candidate for step. In other words, $x_k + \widehat{s}$ is accepted and defined as x_{k+1} as long as there is a sufficient decrease from $f(x_k)$ to $f(x_k + \widehat{s})$. Otherwise, the step $\widehat{s}$ is rejected, the size of set $\mathcal{B}_\Delta$ is decreased by reducing the trust–region radius Δ and a new quadratic subproblem is defined.

As usual in modern trust–region methods, it is not necessary to accurately solve the subproblem to obtain global convergence of the main algorithm (cf. [3, 4, 11]). Instead, a mild condition that relates the target value of the quadratic model for decrease (2) to the solution of a very simple auxiliary subproblem is sufficient to ensure that every accumulation point is stationary.

Distinct strategies for approximately solving the quadratic subproblem and updating the trust–region radius generate distinct algorithms. Moreover, the choice of the norm that defines the trust region is also relevant to the treatment given to the quadratic subproblem. The set $\mathcal{B}_\Delta$ is still box–shaped when the ℓ_∞ norm is used, as in [3, 4, 11]. In this case,

$$\mathcal{B}_\Delta = \{s \in I\!R^n \mid l_k \leq s \leq u_k\} \tag{3}$$

where $(l_k)_i = \max\{l_i - (x_k)_i, -\Delta\}$ and $(u_k)_i = \min\{u_i - (x_k)_i, \Delta\}$, $i = 1, \ldots, n$. Alternatively, $\mathcal{B}_\Delta$ will be the intersection of the original box with an Euclidian ball if the ℓ_2 norm is adopted (cf. [2, 7]).

In (2), the matrix B_k can be the Hessian $\nabla^2 f(x_k)$ in case f is twice continuously differentiable in $\mathcal{B}$, or any quasi–Newton approximation. Besides being symmetric, a boundedness condition on B_k is the only assumption required for achieving global convergence of the main algorithm.

3 The Algorithm `BOX-QUACAN`

In this section we describe the algorithm `BOX-QUACAN`, for finding approximate solutions of the bound constrained minimization problem (1), where some components of the vectors l and u that define the bounds may be infinite. The algorithm `BOX-QUACAN` is of trust–region type and has been fully analysed in [11]. Here we intend to give a brief description of the method, so that the similarities and differences with the box constrained algorithm that is part of the package `LANCELOT` become more evident.

Let $\sigma_1, \sigma_2, \alpha, \Delta_{min}, \theta$ be such that $0 < \sigma_1 \leq \sigma_2 < 1$, $\alpha \in (0,1)$, $\Delta_{min} > 0$ and $\theta \in (0,1]$. In the beginning, we have an arbitrary initial feasible point x_0, an initial symmetric matrix $B_0 \in I\!R^{n\times n}$ (the Hessian approximation), a nonsingular matrix $D_0 \in I\!R^{n\times n}$ (the scaling matrix) and an initial radius $\Delta^0 \geq \Delta_{min}$. The role of the scaling matrix D_k is to allow the possibility of adjusting the size of variables that have widely differing magnitudes. Given a feasible point x_k, square matrices B_k symmetric and D_k nonsingular and $\Delta^k \geq \Delta_{min}$, the steps for obtaining x_{k+1} and Δ_k are given as follows:

Algorithm `BOX-QUACAN`

Step 1. (Set the initial radius of the trust region and compute an upper bound of $\|B_k\|_2$)

Set $\Delta \leftarrow \Delta^k$.

Compute $M_k > 0$ such that $\|B_k\|_2 \leq M_k$.

Step 2. (Solve the "easy" subproblem)

Compute a global solution $s_k^Q(\Delta)$ of

$$\begin{array}{ll} \text{Minimize} & Q_k(s) \equiv \frac{1}{2} M_k s^T s + g_k^T s \\ \text{s.t.} & l \leq x_k + s \leq u \\ & \|D_k s\|_\infty \leq \Delta\,. \end{array} \tag{4}$$

If $Q_k(s_k^Q(\Delta)) = 0$, stop.

Step 3. (Compute the trial step)

Compute $\overline{s}_k(\Delta)$ such that

$$\begin{array}{c} q_k(\overline{s}_k(\Delta)) \leq \theta Q_k(s_k^Q(\Delta)) \\ l \leq x_k + \overline{s}_k(\Delta) \leq u \\ \|D_k \overline{s}_k(\Delta)\|_\infty \leq \Delta \end{array} \tag{5}$$

where $q_k(s) = \frac{1}{2} s^T B_k s + g_k^T s$ for all $s \in I\!R^n$.

Step 4. (Test sufficient decrease)

If $f(x_k + \overline{s}_k(\Delta)) \leq f(x_k) + \alpha q_k(\overline{s}_k(\Delta))$ then

define $s_k = \overline{s}_k(\Delta)$, $x_{k+1} = x_k + s_k$, $\Delta_k = \Delta$ and return.

Else, $\Delta \leftarrow \Delta_{new} \in [\sigma_1 \|D_k \overline{s}_k(\Delta)\|_\infty, \sigma_2 \Delta]$ and repeat Step 2.

Remark: Defining $\widetilde{s} = -g_k/M_k$, then $\|s - \widetilde{s}\|_2^2 = 2Q_k(s)/M_k + \|\widetilde{s}\|_2^2$ and so $s_k^Q(\Delta)$ is the Euclidean projection of $\widetilde{s}$ on the feasible region of (4). Computing this projection is trivial in many practical situations. For example, if D_k is diagonal then the feasible set of (4) is still box–shaped: (3) holds with $(l_k)_i = \max\{l_i - (x_k)_i, -\Delta/(D_k)_{ii}\}$ and $(u_k)_i = \min\{u_i - (x_k)_i, \Delta/(D_k)_{ii}\}$, $i = 1, \ldots, n$.

The following results are proved in [11]. The global convergence result (Theorem 2) is also valid for the bound constrained algorithm of `LANCELOT` and, in fact, it is the type of result that one should expect from every reasonable method for bound constrained minimization.

Lemma 1. *If the algorithm* `BOX-QUACAN` *stops at Step 2 (so $Q_k(s_k^Q(\Delta)) = 0$) then x_k is a Karush–Kuhn–Tucker point of the problem (1).*

Theorem 1. *The algorithm* `BOX-QUACAN` *is well defined, that is, if it does not stop at Step 2 (with $Q_k(s_k^Q(\Delta)) = 0$) then x_{k+1} can be computed repeating Steps 2–4 a finite number of times.*

Theorem 2. *Assume that $\{x_k\}$ is an infinite sequence generated by the algorithm* `BOX- QUACAN`*, $I\!K_1$ is an infinite set of indices such that $\lim_{k \in I\!K_1} x_k = x_*$ and M_k, $\|D_k\|_\infty$ and $\|D_k^{-1}\|_\infty$ are bounded for $k \in I\!K_1$. Then x_* is a stationary (Karush–Kuhn–Tucker) point of (1).*

A Fortran double precision code was written that implements the algorithm `BOX-QUACAN`. The chosen set of parameters is specified in Section 5. We observe that $\overline{s}_k(\Delta)$ satisfying (5) in Step 3 exists, since $s_k^Q(\Delta)$ is an admissible choice. However, in order to improve the performance of the algorithm, for computing $\overline{s}_k(\Delta)$ we considered the following problem:

$$\begin{array}{ll} \text{Minimize} & q_k(s) \\ \text{s.t.} & l \le x_k + s \le u \\ & \|D_k s\|_\infty \le \Delta\,. \end{array} \tag{6}$$

For approximately solving (6) we used an algorithm that minimizes quadratics on a box (cf. [9, 11]). This quadratic minimization solver proceeds combining conjugate gradient iterations and projected gradients, so that only matrix–vector products are required, no matrix factorizations are used and, in consequence, no fill–in is produced. The stopping criterion for the algorithm that solves (6) is

$$\|P[\nabla q_k(s), \mathcal{B}_{\Delta_k}]\|_2 \le \tau \|P[\nabla q_k(0), \mathcal{B}_{\Delta_k}]\|_2\,, \tag{7}$$

where $P[x, \mathcal{S}]$ denotes the projection of x on the set $\mathcal{S}$ and $\tau \in (0,1)$ is the demanded accuracy. Of course, we also required the accomplishment of condition (5). The initial approximation for the quadratic minimization was $s_k^Q(\Delta)$. This choice is reasonable because it guarantees that the condition (5) imposed for convergence will be satisfied.

4 The Package LANCELOT Applied to Bound-constrained Minimization

The package LANCELOT, developed by Conn, Gould and Toint [6] can be applied to general large–scale nonlinear programming problems

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{s.t.} & c(x) = 0 \\ & l \leq x \leq u \end{array} \tag{8}$$

where $f : I\!R^n \to I\!R$ and $c : I\!R^n \to I\!R^m$ are assumed to be differentiable on the set $\mathcal{B}$. Any component of the bounds in (8) may be infinite.

Using augmented Lagrangian techniques, the problem (8) can be solved by a sequence of simple bound constrained problems

$$\begin{array}{ll} \text{Minimize} & \Phi(x, \lambda, S, \mu) \equiv f(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} s_{ii} c_i(x)^2 \\ \text{s.t.} & l \leq x \leq u \end{array} \tag{9}$$

where, for all $i = 1, \ldots, m$, the values λ_i are the components of the Lagrange multiplier estimates vector λ, s_{ii} are the diagonal entries of the positive definite scaling matrix S and μ is the penalty parameter [5].

Thus, the heart of the software is an efficient solver for problem (1), which is applied to (9) successively, generating a sequence of iterates $\{x_k\}$. In LANCELOT, this solver is called SBMIN, an algorithm for finding approximate solutions of the bound–constrained problem (1).

In this section we will devote our attention to the algorithm SBMIN, which is the equivalent in LANCELOT to BOX-QUACAN. Algorithm SBMIN is also of trust–region type, but it differs from BOX-QUACAN in some features that are briefly described below. For more details, see [4, 6].

Let us consider $q_k(s)$ the quadratic model for decrease of f around x_k, for $k = 0, 1, 2, \ldots$ as defined in (2). A trial step $\overline{s}_k$ is determined through the approximate solution of problem

$$\begin{array}{ll} \text{Minimize} & q_k(s) \\ \text{s.t.} & l \leq x_k + s \leq u \\ & \|s\| \leq \Delta_k \, , \end{array} \tag{10}$$

where $\|\cdot\|$ is an adequate norm and Δ_k is the current trust–region radius. As in the algorithm BOX-QUACAN, it is convenient to use the ℓ_∞ norm, so that the feasible region of subproblem (10) becomes the box $\mathcal{B}_{\Delta_k}$ defined in (3). At each iteration of SBMIN the trial step that approximately minimizes the subproblem (10) is obtained after a previous computation of a step that ensures global convergence: the generalized Cauchy step.

The generalized Cauchy step (GCS), denoted by s_k^C, is computed by a projected gradient algorithm. The so–called fraction of Cauchy decrease condition

$$q_k(\overline{s}_k) \leq \theta q_k(s_k^C), \quad \theta \in (0,1)$$

is an important ingredient in the proof of global convergence of algorithm SBMIN.

Considering a positive definite diagonal scaling matrix D_k and $\overline{g}_k = D_k g_k$, the projection of direction $-\overline{g}_k$ in the feasible set $\mathcal{B}_{\Delta_k}$ yields the poligonal line $P[x_k - t\overline{g}_k, \mathcal{B}_{\Delta_k}]$, for $t \geq 0$. A piecewise quadratic function is thus defined by

$$\overline{q}_k(t) = q_k(P[x_k - t\overline{g}_k, \mathcal{B}_{\Delta_k}]), \quad t \geq 0.$$

The value t_k^C corresponding to the first local minimizer of $\overline{q}_k(t)$ defines

$$x_k^C = P[x_k - t_k^C \overline{g}_k, \mathcal{B}_{\Delta_k}],$$

which is the generalized Cauchy point at the k–th iteration and the generalized Cauchy step is given by

$$s_k^C = x_k^C - x_k.$$

Once the step s_k^C is determined, the idea is to achieve a further decrease in the quadratic model by a better trial step that accelerates the convergence of the method. Considering the current feasible set $\mathcal{B}_{\Delta_k}$, the trial step $\overline{s}_k$ is defined so that the active components of s_k^C are kept fixed and the algorithm tries to decrease the reduced quadratic defined in terms of the free variables. This is done by means of the conjugate gradient method using s_k^C as the initial approximation. The algorithm stops with a trial step $\overline{s}_k$ when the norm of the reduced gradient of the model is less than η_k given by

$$\eta_k = \min\{0.1, \sqrt{\|P[\nabla q_k(0), \mathcal{B}_{\Delta_k}]\|_\infty}\} \|P[\nabla q_k(0), \mathcal{B}_{\Delta_k}]\|_\infty$$

or when any of the free variables violates a bound of $\mathcal{B}_{\Delta_k}$.

The algorithm SBMIN is described as follows.

Let γ_0, γ_1, γ_2, α and η be such that $0 < \gamma_0 \leq \gamma_1 < 1 < \gamma_2$, $\alpha > 0$ and $\eta > 0$. The feasible starting point x_0, the initial trust–region radius Δ_0 and the initial symmetric approximation to the Hessian matrix B_0 are given. At each iteration k, the steps to obtain x_{k+1} and Δ_{k+1} are given in the following:

Algorithm SBMIN

Step 1. (Test for convergence)

Compute the projected gradient $(g_k)_P \equiv P[x_k - g_k, \mathcal{B}_{\Delta_k}] - x_k$.

If $\|(g_k)_P\|_\infty = 0$, stop.

Step 2. (Compute the GCS)

Calculate the bounds l_k and u_k of the current feasible set $\mathcal{B}_{\Delta_k}$ defined in (3) and obtain the GSC s_k^C as described previously.

Step 3. (Compute the trial step)

Apply the conjugate gradient algorithm starting from s_k^C to find an approximation $\overline{s}_k$ to the minimizer of the model quadratic (2) over the feasible region (3), with the additional restriction that the variables that are on their bounds in s_k^C remain fixed.

Step 4. (Test sufficient decrease and update current point and current trust–region radius)

Compute $\rho_k = \dfrac{f(x_k+\overline{s}_k)-f(x_k)}{q_k(\overline{s}_k)}$.

Set

$$x_{k+1} = \begin{cases} x_k + \overline{s}_k & \text{if} \quad \rho_k > \alpha\,, \\ x_k & \text{if} \quad \rho_k \le \alpha \end{cases}$$

and

$$\Delta_{k+1} = \begin{cases} \gamma_0^{(k)}\Delta_k & \text{if} \quad \rho_k \le \alpha\,, \\ \Delta_k & \text{if} \quad \alpha < \rho_k < \eta\,, \\ \gamma_2^{(k)}\Delta_k & \text{if} \quad \rho_k \ge \eta\,, \end{cases}$$

where

$$\gamma_0^{(k)} = \begin{cases} \max\left\{\gamma_0, \gamma_1 \frac{\|\overline{s}_k\|}{\Delta_k}\right\} & \text{if} \quad 0 \le \rho_k \le \alpha\,, \\ \\ \max\left\{\gamma_0, \frac{(1-\eta)g_k^T\|\overline{s}_k\|}{f(x_k)-f(x_k+\overline{s}_k)+(1-\eta)g_k^T\overline{s}_k+\eta q_k(\overline{s}_k)}\right\} & \text{if} \quad \rho_k < 0 \end{cases}$$

and

$$\gamma_2^{(k)} = \max\left\{1, \gamma_2\frac{\|\overline{s}_k\|}{\Delta_k}\right\}\,.$$

It is worthwhile noticing that it might be that x_{k+1} coincides with x_k, what guarantees that the algorithm is well defined. Naturally, it is only when the iterate has changed that it is necessary to recompute the gradient and a new second derivative approximation. The convergence results of the algorithm `SBMIN` are proved in [3].

5 Numerical Results

The comparison of the numerical performance of the algorithms `BOX-QUACAN` and `LANCELOT` was made through the resolution of 220 problems selected from the `CUTE` collection (version of 1993). Several sets of problems can be obtained from this collection, according to given specified features, such as type of objective function, type of constraints, origin of problem, etc. With the aim not contaminating the final results of the comparison by any particular feature of the considered problems that might be more favorable to one approach than to the other, in our numerical tests we selected problems with several but well specified features. In this sense, the 220 test problems are distributed in fifteen sets, according to type of objective function (quadratic (Q), least squares (L) or any other nonlinear type (O)); type of constraints (unconstrained (U) or simple bounded (B)) and origin of problem (academic (A), i.e. that has been constructed specifically by researchers to test one or more algorithms, real (R), i.e. the problem solution has been used in a real application for purposes other than testing algorithms and modelling (M), i.e. the problem is part of a modelling exercise where the actual value of the solution is not used in a genuine practical application). Among the eighteen sets that could be generated by combining the features above, imposing existence of second derivatives and regularity, the selection process of `CUTE` generated only fifteen sets: the combinations real unconstrained least–squares, other real unconstrained and other real bound–constrained did not produce any problem.

Since the choices of parameters for `BOX-QUACAN` and `SBMIN` are related to the different set of test problems, first we will specify the set of selected problems of our comparative tests:

- Academic unconstrained quadratic problems (AUQ)
- Academic unconstrained least–squares problems (AUL)
- Other academic unconstrained problems (OAU)
- Academic bound–constrained quadratic problems (ABQ)
- Academic bound–constrained least–squares problems (ABL)
- Other academic bound–constrained problems (OAB)
- Real unconstrained quadratic problems (RUQ)
- Real bound–constrained quadratic problems (RBQ)
- Real bound–constrained least–squares problems (RBL)
- Modelling unconstrained quadratic problems (MUQ)
- Modelling unconstrained least–squares problems (MUL)
- Other modelling unconstrained problems (OMU)
- Modelling bound–constrained quadratic problems (MBQ)
- Modelling bound–constrained least–squares problems (MBL)
- Other modelling bound–constrained problems (OMB)

The tests were developed in Fortran 77 double precision and run in a SUN Sparc Station 2 with the `-O` compiler option for both codes. An interface was built for running `BOX-QUACAN` with the `CUTE` collection which decodifies the Standard Input Format (SIF) to produce Fortran subroutines for evaluating the objective function, its gradient and the product of its Hessian by a vector [12]. In the following we specify the chosen parameters so that both algorithms become as closely comparable as possible.

In the algorithm BOX-QUACAN we used $\sigma_1 = \sigma_2 = 0.5$, $\alpha = 0.1$, $\Delta_{min} = 10^{-4}$, $\theta = 10^{-3}$, $B_k = \nabla^2 f(x_k)$ and $D_k = I$ for all k. For non-quadratic problems the initial trust–region radius was chosen according to the following. We computed

$$\Delta_{max} = \min\{10^5, \|u - l\|_\infty\}$$

and

$$\xi = \|P[g_0, \mathcal{B}]\|_2 \frac{\max\{1, \|x_0\|_2\}}{\max\{1, |f(x_0)|\}}.$$

Then,

$$\begin{array}{llll} \text{if} & \xi < 0.5 & \text{we set} & \Delta^0 = \min\{0.1\Delta_{max}, 10\}; \\ \text{if} & 0.5 \leq \xi < 10 & \text{we set} & \Delta^0 = \min\{0.5\Delta_{max}, 100\}; \\ \text{if} & \xi \geq 10 & \text{we set} & \Delta^0 = \min\{\Delta_{max}, 1000\}. \end{array}$$

For quadratic problems we set $\Delta^0 = \Delta_{max}$.

A point is declared stationary whenever $\|(g_k)_P\|_\infty \leq \varepsilon_g$ for non-quadratic problems ($(g_k)_P$ defined by Step 1 of Algorithm SBMIN) or $\|P[g_k, \mathcal{B}]\|_2 \leq \varepsilon_g$ for quadratic problems, with tolerance for norm of projected gradient $\varepsilon_g = 10^{-5}$. Other reasons for stopping are that 1000 functional evaluations are performed or that the trust–region radius becomes too small ($\Delta_k \leq \varepsilon_\Delta$), with tolerance $\varepsilon_\Delta = 10^{-8}$. In the stopping criterion (7) we used $\tau = 0.1$ and $\tau = \varepsilon_g / \|P[\nabla q_k(0), \mathcal{B}_{\Delta_k}]\|_2$, respectively for non-quadratic and quadratic problems.

It is important to stress that with the choices specifically made for the quadratic problems we intended to ensure that in this case the trust–region algorithm works just as a main program that interfaces the problem data and the quadratic solver.

For most of the problems, the upper bound M_k was set to 10^5. In a few cases (marked with # in Tables 1-15) we used $M_k = 10^{10}$.

We run the algorithm SBMIN with the following choices:

- exact-second-derivatives-used
- cg-method-used
- exact-Cauchy-point-required
- infinity-norm-trust-region-used
- gradient-tolerance 10^{-5} (i.e. convergence is declared if $\|(g_k)_P\|_\infty \leq 10^{-5}$)
- maximum-number-of-iterations 1000

The initial trust region was the default value $\Delta_0 = 0.1\|(g_0)_P\|_\infty$, with $(g_0)_P$ as defined in Step 1 of Algorithm SBMIN. The other parameters of algorithm SBMIN were $\gamma_0 = 0.0625$, $\gamma_1 = 0.5$, $\gamma_2 = 2$, $\alpha = 0.25$, $\eta = 0.75$ and $\varepsilon_\Delta = 10^{-12}$.

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, 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.

In Tables 1-15 the following notation is used: N is the dimension of the problem; S identifies the software used (B for BOX-QUACAN and L for LANCELOT); RS is the reason for stopping: g_p indicates that the projected gradient is sufficiently small, ∞_F indicates that the maximum allowed number of functional evaluations was reached, ∞_I indicates that the maximum allowed number of outer iterations was reached and Δ indicates that an excessively small trust–region radius was computed. The pair (IT_{out}, IT_{inn}) contains the number of outer and inner iterations (quadratic solver) of each algorithm. The pair (FE, GE) informs the number of functional and gradient evaluations; $f(x)$ and $\|g_p\|$ are respectively the objective function value and the ℓ_∞ norm of the projected gradient at the final point. The value T gives the CPU time in seconds spent by each test.

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
DIXON3DQ	10	B	g_p	(1, 10)	(2,1)	0.9D−29	0.6D−14	0.01
		L	g_p	(4, 22)	(4, 5)	0.3D−30	0.9D−15	0.06
DQDRTIC	5000	B	g_p	(1, 5)	(2, 1)	0.2D−19	0.6D−10	2.25
		L	g_p	(2, 9)	(2, 3)	0.1D−29	0.2D−14	3.90
HILBERTA	10	B	g_p	(1, 5)	(2, 1)	0.1D−08	0.2D−05	0.02
		L	g_p	(4, 7)	(4, 5)	0.2D−06	0.5D−05	0.10
HILBERTB	50	B	g_p	(1, 4)	(2, 1)	0.7D−12	0.2D−05	0.18
		L	g_p	(3, 3)	(3, 4)	0.8D−12	0.1D−05	0.50
TESTQUAD	1000	B	g_p	(1, 707)	(2, 1)	0.2D−11	0.2D−05	23.50
		L	g_p	(3, 787)	(3, 4)	0.1D−11	0.3D−05	16.70
TRIDIA	10000	B	g_p	(1, 1095)	(2, 1)	0.1D−13	0.2D−05	371.00
		L	g_p	(3, 1261)	(3, 4)	0.3D−13	0.1D−05	294.60
ZANGWIL2	2	B	g_p	(1, 1)	(2, 1)	−0.2D+02	0.0D+00	0.01
		L	g_p	(3, 0)	(3, 4)	−0.2D+00	0.0D+00	0.04

Table 1: Academic unconstrained quadratic problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
BARD	3	B	g_p	(12, 29)	(14, 12)	0.8D−02	0.6D−06	0.05
		L	g_p	(11, 28)	(11, 11)	0.8D−02	0.2D−05	0.10
BDQRTIC	1000	B	g_p	(14, 74)	(15, 14)	0.4D+04	0.3D−05	8.04
		L	g_p	(10, 80)	(10, 11)	0.4D+04	0.2D−05	7.80
BEALE	2	B	g_p	(7, 22)	(16, 7)	0.2D−16	0.2D−07	0.04
		L	g_p	(11, 17)	(11, 11)	0.3D−16	0.5D−08	0.10
BIGGS6	6	B	g_p	(25, 143)	(44, 25)	0.6D−02	0.9D−06	0.25
		L	g_p	(23, 72)	(23, 21)	0.2D−06	0.4D−05	0.30
BOX3	3	B	g_p	(8, 16)	(9, 8)	0.2D−14	0.2D−08	0.02
		L	g_p	(7, 14)	(7, 8)	0.2D−08	0.2D−05	0.10
BROWNAL	10	B	g_p	(7, 10)	(8, 7)	0.2D−09	0.3D−05	0.03
		L	g_p	(4, 6)	(4, 5)	0.1D−09	0.8D−05	0.10
BROWNBS	2	B#	g_p	(112, 314)	(113, 112)	0.0D+00	0.0D+00	0.23
		L	g_p	(6, 7)	(6, 7)	0.2D−30	0.9D−09	0.05
BROWNDEN	4	B	g_p	(10, 30)	(11, 10)	0.9D+05	0.7D−10	0.04
		L	g_p	(8, 30)	(8, 9)	0.9D+05	0.3D−09	0.10
BRYBND	5000	B	g_p	(9, 68)	(10, 9)	0.5D−11	0.4D−05	39.25
		L	g_p	(20, 187)	(20, 17)	0.7D−12	0.5D−05	108.90
CHNROSNB	50	B	g_p	(43, 682)	(71, 43)	0.1D−10	0.3D−05	1.54
		L	g_p	(85, 598)	(85, 67)	0.1D−12	0.3D−05	1.90
CUBE	2	B	g_p	(34, 80)	(58, 34)	0.3D−10	0.4D−05	0.05
		L	g_p	(47, 80)	(47, 40)	0.3D−12	0.3D−06	0.10
DENSCHNB	2	B	g_p	(3, 14)	(11, 3)	0.0D+00	0.0D+00	0.02
		L	g_p	(6, 4)	(6, 7)	0.3D−11	0.4D−05	0.10

Table 2: Academic unconstrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\Vert g_p \Vert$	T
DENSCHNC	2	B	g_p	(9, 13)	(10, 9)	0.2D−12	0.1D−05	0.03
		L	g_p	(13, 21)	(13, 12)	0.2D−19	0.8D−09	0.10
DENSCHND	3	B	g_p	(29, 84)	(39, 29)	0.1D−07	0.7D−05	0.06
		L	g_p	(32, 68)	(32, 29)	0.1D−06	0.4D−05	0.30
DENSCHNE	3	B	g_p	(13, 27)	(30, 13)	0.5D−16	0.1D−07	0.01
		L	g_p	(15, 13)	(15, 15)	0.3D−11	0.3D−05	0.10
DENSCHNF	2	B	g_p	(6, 12)	(7, 6)	0.7D−21	0.6D−09	0.01
		L	g_p	(6, 12)	(6, 7)	0.7D−21	0.6D−09	0.10
EIGENALS	110	B	g_p	(20, 137)	(29, 20)	0.1D−08	0.7D−05	3.51
		L	g_p	(20, 138)	(20, 19)	0.2D−09	0.4D−05	3.10
EIGENBLS	110	B	g_p	(74, 5607)	(155, 74)	0.9D−10	0.3D−05	144.08
		L	g_p	(147, 3228)	(147, 126)	0.3D−09	0.3D−05	83.50
EIGENCLS	462	B	g_p	(83, 9507)	(165, 83)	0.1D−09	0.3D−05	2118.36
		L	g_p	(181, 3448)	(181, 145)	0.3D−10	0.4D−05	768.00
ENGVAL2	3	B	g_p	(19, 40)	(27, 19)	0.4D−14	0.6D−07	0.04
		L	g_p	(20, 32)	(20, 19)	0.9D−12	0.1D−05	0.20
ERRINROS	50	B	g_p	(60, 902)	(106, 60)	0.4D+02	0.2D−05	2.21
		L	g_p	(60, 677)	(60, 54)	0.4D+02	0.2D−05	2.00
EXPFIT	2	B	g_p	(6, 12)	(17, 6)	0.2D+00	0.5D−08	0.03
		L	g_p	(10, 12)	(10, 9)	0.2D+00	0.2D−08	0.10
EXTROSNB	10	B	∞_F	(506, 5646)	(1000, 506)	0.5D−06	0.2D−03	3.69
		L	g_p	(394, 3024)	(394, 327)	0.2D−05	0.8D−05	3.60
FREUROTH	5000	B	g_p	(12, 54)	(28, 12)	0.6D+06	0.3D−06	32.41
		L	g_p	(15, 33)	(15, 13)	0.6D+06	0.3D−06	22.50
GENROSE	500	B	g_p	(226, 6283)	(450, 226)	0.1D+01	0.1D−05	133.71
		L	g_p	(527, 2994)	(527, 460)	0.1D+01	0.2D−05	80.00
GROWTHLS	3	B	g_p	(136, 508)	(275, 136)	0.1D+01	0.9D−07	0.86
		L	g_p	(187, 471)	(187, 160)	0.1D+01	0.1D−06	1.60
HATFLDD	3	B	g_p	(19, 52)	(28, 19)	0.7D−07	0.4D−06	0.08
		L	g_p	(19, 33)	(19, 19)	0.8D−07	0.6D−05	0.10
HATFLDE	3	B	g_p	(17, 44)	(25, 17)	0.5D−06	0.4D−05	0.10
		L	g_p	(20, 39)	(20, 20)	0.5D−06	0.3D−05	0.20
HELIX	3	B	g_p	(13, 37)	(23, 13)	0.1D−10	0.5D−05	0.05
		L	g_p	(14, 29)	(14, 11)	0.3D−14	0.7D−05	0.10
HIMMELBF	4	B	g_p	(57, 360)	(121, 57)	0.3D+03	0.4D−05	0.28
		L	g_p	(299, 996)	(299, 262)	0.3D+03	0.2D−05	2.60
HYDC20LS	99	B#	∞_F	(993, 20005)	(1000, 993)	0.7D+01	0.2D+01	262.19
		L	∞_I	(1000, 212657)	(999,990)	0.7D+00	0.2D+04	2421.10
LIARWHD	10000	B	g_p	(15, 24)	(18, 15)	0.5D−09	0.1D−05	39.33
		L	g_p	(12, 22)	(12, 13)	0.1D−19	0.3D−07	26.90
MANCINO	100	B	g_p	(13, 40)	(14, 13)	0.9D−17	0.2D−05	157.68
		L	g_p	(11, 22)	(11, 7)	0.2D−20	0.5D−07	41.20
MSQRTALS	1024	B	g_p	(29, 6862)	(56, 29)	0.5D−08	0.8D−05	4134.11
		L	g_p	(38, 5021)	(38, 29)	0.7D−09	0.8D−06	3175.00
MSQRTBLS	1024	B	g_p	(48, 13266)	(95, 48)	0.3D−09	0.2D−05	7397.93
		L	g_p	(34, 3544)	(34, 28)	0.2D−09	0.1D−05	2744.30
NONDIA	10000	B	g_p	(4, 5)	(5, 4)	0.6D−09	0.2D−06	8.45
		L	g_p	(4, 4)	(4, 5)	0.4D−15	0.3D−05	8.10
NONMSQRT	1024	B	∞_F	(969, 51230)	(1000, 969)	0.9D+02	0.3D−02	26497.87
		L	∞_I	(1000, 456057)	(1000, 910)	0.9D+02	0.4D+00	229492.00
PENALTY1	1000	B	g_p	(39, 177)	(49, 39)	0.1D-02	0.9D−05	15.05
		L	g_p	(49, 54)	(49, 39)	0.1D-02	0.7D−05	7.60
PENALTY2	100	B	g_p	(19, 80)	(20, 19)	0.1D+06	0.2D−05	0.95
		L	g_p	(19, 575)	(19, 20)	0.1D+06	0.2D−05	3.70
PFIT1LS	3	B	g_p	(430, 2191)	(868, 430)	0.4D−06	0.9D−05	1.91
		L	g_p	(535, 1069)	(535, 466)	0.6D−07	0.1D−04	2.90

Table 2 (cont.): Academic unconstrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
PFIT2LS	3	B	g_p	(295, 1750)	(597, 295)	0.4D−07	0.7D−05	1.33
		L	g_p	(207, 432)	(207, 178)	0.7D−07	0.1D−04	1.20
PFIT3LS	3	B	g_p	(106, 535)	(217, 106)	0.1D−07	0.7D−05	0.48
		L	g_p	(181, 349)	(181, 155)	0.1D−07	0.9D−05	1.00
PFIT4LS	3	B	g_p	(172, 680)	(350, 172)	0.1D−07	0.9D−05	0.69
		L	g_p	(397, 790)	(397, 354)	0.1D−07	0.9D−05	2.30
ROSENBR	2	B	g_p	(25, 61)	(48, 25)	0.4D−13	0.8D−06	0.02
		L	g_p	(31, 55)	(31, 27)	0.4D−10	0.5D−05	0.20
S308	2	B	g_p	(9, 15)	(10, 9)	0.8D+00	0.9D−05	0.02
		L	g_p	(9, 14)	(9, 10)	0.8D+00	0.3D−07	0.10
SENSORS	100	B	g_p	(9, 51)	(19, 9)	−0.2D+04	0.2D−06	44.34
		L	g_p	(14, 20)	(14, 10)	−0.2D+04	0.2D−06	25.30
SINEVAL	2	B	g_p	(59, 173)	(117, 59)	0.3D−14	0.3D−07	0.15
		L	g_p	(69, 100)	(69, 62)	0.1D−12	0.2D−06	0.30
SPMSRTLS	10000	B	g_p	(200, 10580)	(263, 200)	0.3D+00	0.7D−05	7969.27
		L	g_p	(17, 275)	(17, 14)	0.5D−10	0.1D−05	226.20
SROSENBR	10000	B	g_p	(7, 9)	(8, 7)	0.4D−07	0.2D−05	9.56
		L	g_p	(12, 22)	(12, 10)	0.4D−10	0.7D−07	18.80
TQUARTIC	10000	B	g_p	(7, 13)	(12, 7)	0.7D−04	0.2D−05	15.08
		L	g_p	(9, 8)	(9, 10)	0.3D−10	0.1D−05	36.80
VAREIGVL	5000	B	g_p	(14, 64)	(15, 14)	0.1D−07	0.1D−05	28.46
		L	g_p	(13, 123)	(13, 14)	0.8D−07	0.3D−05	46.00
WATSON	31	B	g_p	(10, 47)	(11, 10)	0.1D−07	0.4D−06	0.46
		L	g_p	(9, 46)	(9, 10)	0.3D−07	0.3D−05	0.60
WOODS	10000	B	g_p	(17, 38)	(18, 17)	0.2D−07	0.6D−05	31.35
		L	g_p	(16, 59)	(16, 17)	0.7D−09	0.1D−04	37.50

Table 2 (cont.): Academic unconstrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
ALLINITU	4	B	g_p	(9, 31)	(20, 9)	0.5D+01	0.3D−05	0.04
		L	g_p	(13, 15)	(13, 12)	0.5D+01	0.1D−06	0.10
ARWHEAD	5000	B	g_p	(5, 6)	(6, 5)	0.1D−11	0.1D−05	6.61
		L	g_p	(5, 4)	(5, 6)	0.0D+00	0.3D−06	10.90
BRKMCC	2	B	g_p	(3, 5)	(4, 3)	0.2D+00	0.1D−05	0.01
		L	g_p	(4, 5)	(4, 5)	0.2D+00	0.2D−10	0.04
BROYDN7D	1000	B	g_p	(13, 434)	(26, 13)	0.4D+03	0.4D−05	37.45
		L	g_p	(125, 381)	(125, 98)	0.4D+03	0.1D−05	40.90
CLIFF	2	B#	g_p	(27, 28)	(28, 27)	0.2D+00	0.6D−07	0.02
		L	g_p	(27, 4)	(27, 28)	0.2D+00	0.4D−05	0.10
CRAGGLVY	5000	B	g_p	(16, 118)	(17, 16)	0.2D+04	0.6D−05	58.34
		L	g_p	(12, 327)	(12, 13)	0.2D+04	0.6D−06	91.1
DENSCHNA	2	B	g_p	(5, 9)	(6, 5)	0.1D−14	0.7D−07	0.01
		L	g_p	(5, 6)	(5, 6)	0.7D−13	0.5D−06	0.05
DIXMAANA	3000	B	g_p	(8, 25)	(15, 8)	0.1D+01	0.1D−05	10.11
		L	g_p	(9, 12)	(9, 8)	0.1D+01	0.4D−08	5.70
DIXMAANB	3000	B	g_p	(7, 11)	(8, 7)	0.1D+01	0.9D−05	6.06
		L	g_p	(14, 32)	(14, 12)	0.1D+01	0.1D−05	11.50
DIXMAANC	3000	B	g_p	(9, 13)	(10, 9)	0.1D+01	0.4D−05	7.69
		L	g_p	(16, 40)	(16, 14)	0.1D+01	0.7D−06	14.80
DIXMAAND	3000	B	g_p	(10, 16)	(11, 10)	0.1D+01	0.2D−05	8.70
		L	g_p	(10, 26)	(10, 9)	0.1D+01	0.2D−08	8.20
DIXMAANE	3000	B	g_p	(10, 367)	(20, 10)	0.1D+01	0.9D−06	70.11
		L	g_p	(12, 291)	(12, 11)	0.1D+01	0.1D−05	37.80
DIXMAANF	3000	B	g_p	(12, 213)	(13, 12)	0.1D+01	0.3D−05	43.25
		L	g_p	(22, 287)	(22, 20)	0.1D+01	0.2D−05	46.90

Table 3: Other academic unconstrained problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
DIXMAANG	3000	B	g_p	(14, 771)	(32, 14)	0.1D+01	0.9D−05	142.44
		L	g_p	(20, 260)	(20, 17)	0.1D+01	0.9D−05	41.50
DIXMAANH	3000	B	g_p	(15, 912)	(33, 15)	0.1D+01	0.3D−05	162.97
		L	g_p	(21, 293)	(21, 18)	0.1D+01	0.9D−0.5	47.00
DIXMAANI	3000	B	g_p	(15, 5941)	(31, 15)	0.1D+01	0.3D−05	889.85
		L	g_p	(10, 5758)	(10, 9)	0.1D+01	0.2D−05	615.20
DIXMAANJ	3000	B	g_p	(13, 151)	(14, 13)	0.1D+01	0.9D−05	33.08
		L	g_p	(44, 3380)	(44, 36)	0.1D+01	0.7D−05	454.40
DIXMAANK	3000	B	g_p	(14, 164)	(15, 14)	0.1D+01	0.6D−05	35.69
		L	g_p	(38, 2835)	(38, 31)	0.1D+01	0.6D−05	377.20
DIXMAANL	3000	B	g_p	(15, 175)	(16, 15)	0.1D+01	0.4D−05	37.49
		L	g_p	(39, 3187)	(39, 33)	0.1D+01	0.9D−05	408.00
DJTL	2	B#	g_p	(26, 93)	(58, 26)	−0.9D+04	0.3D−06	0.11
		L	g_p	(54, 45)	(54, 45)	−0.9D+04	0.1D−06	0.30
DQRTIC	5000	B	g_p	(40, 164)	(41, 40)	0.1D−05	0.8D−05	46.00
		L	g_p	(2, 9)	(2, 3)	0.1D−28	0.2D−14	3.90
EDENSCH	2000	B	g_p	(15, 33)	(17, 15)	0.1D+05	0.2D−05	10.98
		L	g_p	(12, 79)	(12, 13)	0.1D+05	0.5D−05	12.00
EG2	1000	B	g_p	(3, 3)	(4, 3)	−0.1D+04	0.6D−08	0.70
		L	g_p	(3, 0)	(3, 4)	−0.1D+04	0.6D−08	0.50
ENGVAL1	5000	B	g_p	(10, 24)	(11, 10)	0.5D+04	0.3D−05	16.22
		L	g_p	(7, 35)	(7, 8)	0.5D+04	0.4D−05	13.90
FLETCBV2	10000	B	g_p	(0, 0)	(1, 0)	−0.5D+00	0.2D−07	1.68
		L	g_p	(0, 0)	(0, 1)	−0.5D+00	0.2D−07	1.80
FLETCBV3	10000	B	∞_F	(500, 4445)	(1001, 500)	−0.4D+09	0.3D+01	4342.48
		L	∞_I	(999, 348)	(999, 872)	−0.4D+09	0.3D+01	4058.90
FLETCHBV	100	B	∞_F	(500, 4416)	(1001, 500)	−0.4D+11	0.3D+05	33.82
		L	∞_I	(999, 854)	(999, 859)	−0.6D+11	0.3D+05	40.80
FLETCHCR	100	B	g_p	(157, 1793)	(187, 157)	0.1D−12	0.7D−05	10.22
		L	g_p	(229, 1753)	(229, 195)	0.2D−11	0.8D−05	9.20
HAIRY	2	B	g_p	(22, 50)	(48, 22)	0.2D+02	0.1D−14	0.03
		L	g_p	(90, 85)	(90, 73)	0.2D+02	0.1D−07	0.50
HIMMELBB	2	B	g_p	(20, 20)	(21, 20)	0.1D−07	0.6D−05	0.04
		L	g_p	(23, 6)	(23, 20)	0.1D−11	0.4D−06	0.10
HIMMELBG	2	B	g_p	(5, 9)	(14, 5)	0.8D−13	0.7D−06	0.02
		L	g_p	(5, 3)	(5, 6)	0.2D−14	0.1D−06	0.03
HIMMELBH	2	B	g_p	(5, 13)	(12, 5)	−0.1D+01	0.3D−09	0.01
		L	g_p	(5, 0)	(5, 6)	−0.1D+01	0.7D−10	0.04
INDEF	50	B	∞_F	(991, 1979)	(1000, 991)	−0.5D+09	0.1D+01	13.26
		L	∞_I	(999, 927)	(999, 557)	−0.8D+05	0.1D+02	17.90
JENSMP	2	B	g_p	(9, 14)	(10, 9)	0.1D+03	0.2D−05	0.03
		L	g_p	(9, 18)	(9, 10)	0.1D+03	0.4D−05	0.10
LOGROS	2	B	g_p	(49, 131)	(95, 49)	0.0D+00	0.7D−08	0.10
		L	g_p	(54, 64)	(54, 47)	0.0D+00	0.2D−08	0.20
NCB20B	1000	B	g_p	(15, 3070)	(31, 15)	0.2D+04	0.6D−06	1504.87
		L	g_p	(34, 1497)	(34, 27)	0.2D+04	0.3D−05	662.40
NONDQUAR	10000	B	g_p	(55, 2153)	(99, 55)	0.6D−0.5	0.8D−05	918.42
		L	g_p	(33, 983)	(33, 28)	0.8D−05	0.5D−05	283.50
PENALTY3	100	B	Δ	(24, 160)	(90, 24)	0.1D−02	0.5D−04	171.24
		L	Δ	(31, 104)	(31, 18)	0.1D−02	0.6D−03	98.30
POWELLSG	10000	B	g_p	(15, 55)	(16, 15)	0.3D−04	0.7D−05	35.06
		L	g_p	(15, 61)	(15, 16)	0.1D−04	0.4D−05	29.00
POWER	1000	B#	g_p	(29, 347)	(30, 29)	0.3D−07	0.9D−05	10.16
		L	g_p	(28, 2987)	(28, 29)	0.5D−08	0.3D−05	81.00
QUARTC	10000	B#	g_p	(42, 92)	(43, 42)	0.7D−05	0.4D−05	59.98
		L	g_p	(28, 2756)	(28, 29)	0.4D−05	0.7D−05	58.90

Table 3 (cont.): Other academic unconstrained problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
SCHMVETT	10000	B	g_p	(7, 47)	(8, 7)	−0.3D+05	0.1D−05	63.04
		L	g_p	(3, 55)	(3, 4)	−0.3D+05	0.1D−06	46.60
SINQUAD	10000	B	g_p	(123, 438)	(239, 123)	0.1D−04	0.4D−05	694.05
		L	g_p	(122, 178)	(122, 107)	0.8D−04	0.3D−05	385.00
SISSER	2	B	g_p	(12, 12)	(13, 12)	0.1D−07	0.4D−05	0.01
		L	g_p	(12, 5)	(12, 13)	0.1D−07	0.5D−05	0.10
SNAIL	2	B	g_p	(69, 174)	(116, 69)	0.6D−13	0.5D−06	0.11
		L	g_p	(87, 154)	(87, 78)	0.1D−10	0.6D−05	0.40
TOINTGSS	10000	B	g_p	(3, 3)	(4, 3)	0.1D+02	0.3D−05	10.70
		L	g_p	(3, 12)	(3, 4)	0.1D+02	0.2D−06	11.00
TOINTPSP	50	B	g_p	(14, 151)	(26, 14)	0.2D+03	0.2D−05	0.47
		L	g_p	(33, 91)	(33, 27)	0.2D+03	0.1D−05	0.60
VARDIM	100	B#	g_p	(21, 32)	(31, 21)	0.1D−22	0.7D−09	0.27
		L	g_p	(25, 0)	(25, 26)	0.5D−25	0.4D−10	0.40

Table 3 (cont.): Other academic unconstrained problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
BIGGSB1	1000	B	g_p	(1, 3545)	(2, 1)	0.2D−01	0.9D−05	71.89
		L	g_p	(502, 66509)	(502, 503)	0.2D−01	0.6D−05	1095.90
BQP1VAR	1	B	g_p	(1, 1)	(2, 1)	0.0D+00	0.0D+00	0.01
		L	g_p	(2, 0)	(2, 3)	0.0D+00	0.0D+00	0.03
BQPGABIM	50	B	g_p	(1, 16)	(2, 1)	−0.4D−04	0.3D−05	0.05
		L	g_p	(3, 19)	(3, 4)	−0.4D−04	0.6D−05	0.10
BQPGASIM	50	B	g_p	(1, 17)	(2, 1)	−0.6D−04	0.3D−05	0.06
		L	g_p	(3, 18)	(3, 4)	−0.6D−04	0.2D−05	0.10
BQPGAUSS	2003	B	g_p	(1, 7442)	(2, 1)	−0.4D+00	0.9D−05	983.69
		L	g_p	(7, 9511)	(7, 8)	−0.4D+00	0.1D−05	1174.70
CHENHARK	1000	B	g_p	(1, 3730)	(2, 1)	−0.2D+01	0.7D−06	76.48
		L	g_p	(121, 136276)	(121, 122)	−0.2D+01	0.3D−06	2288.40
HARKERP2	100	B	g_p	(1, 9)	(2, 1)	−0.5D+00	0.1D−11	0.33
		L	g_p	(1, 2)	(2, 2)	−0.5D+00	0.8D−12	0.70
HS3	2	B	g_p	(1, 2)	(2, 1)	0.0D+00	0.0D+00	0.01
		L	g_p	(7, 0)	(7, 8)	0.2D−35	0.0D+00	0.01
HS3MOD	2	B	g_p	(1, 3)	(2, 1)	0.0D+00	0.0D+00	0.01
		L	g_p	(4, 0)	(4, 5)	0.0D+00	0.0D+00	0.04
JNLBRNG1	15625	B	g_p	(1, 1142)	(2, 1)	−0.2D+00	0.9D−05	2176.15
		L	g_p	(24, 2556)	(24, 25)	−0.2D+00	0.4D−05	2170.60
JNLBRNG2	15625	B	g_p	(1, 1106)	(2, 1)	−0.4D+01	0.9D−05	1321.84
		L	g_p	(14, 2673)	(14, 15)	−0.4D+01	0.6D−06	2121.30
JNLBRNGA	15625	B	g_p	(1, 1179)	(2, 1)	−0.3D+00	0.8D−05	1519.47
		L	g_p	(21, 2135)	(21, 22)	−0.3D+00	0.3D−06	1758.30
JNLBRNGB	15625	B	g_p	(1, 2661)	(2, 1)	−0.6D+01	0.9D−05	2334.17
		L	g_p	(11, 4439)	(11, 12)	−0.6D+01	0.9D−06	3295.20
NOBNDTOR	14884	B	g_p	(1, 884)	(2, 1)	−04D+00	0.9D−05	1340.68
		L	g_p	(36, 1539)	(36, 37)	−0.4D+00	0.2D−05	1519.30
OBSTCLAE	15625	B	g_p	(1, 759)	(2, 1)	0.2D+01	0.7D−05	1598.07
		L	g_p	(4, 7608)	(4, 5)	0.2D+01	0.1D−05	6978.70
OBSTCLAL	15625	B	g_p	(1, 305)	(2, 1)	0.2D+01	0.7D−05	415.42
		L	g_p	(24, 805)	(24, 25)	0.2D+01	0.3D−05	579.60

Table 4: Academic bound–constrained quadratic problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
OBSTCLBL	15625	B	g_p	(1, 366)	(2, 1)	0.7D+01	0.9D−05	896.07
		L	g_p	(18, 3259)	(18, 19)	0.7D+01	0.6D−06	2696.50
OBSTCLBM	15625	B	g_p	(1, 242)	(2, 1)	0.7D+01	0.9D−05	477.19
		L	g_p	(5, 1483)	(5, 6)	0.7D+01	0.2D−05	1372.90
OBSTCLBU	15625	B	g_p	(1, 443)	(2, 1)	0.7D+01	0.8D−05	762.01
		L	g_p	(19, 1102)	(19, 20)	0.7D+01	0.2D−05	923.00
OSLBQP	8	B	g_p	(1, 1)	(2, 1)	0.6D+01	0.0D+00	0.01
		L	g_p	(2, 0)	(2, 3)	0.6D+01	0.0D+00	0.04
QUDLIN	12	B	g_p	(1, 1)	(2, 1)	−0.7D+04	0.0D+00	0.02
		L	g_p	(1, 0)	(1, 2)	−0.7D+04	0.0D+00	0.06
SIM2BQP	2	B	g_p	(1, 1)	(2, 1)	0.0D+00	0.0D+00	0.01
		L	g_p	(2, 0)	(2, 3)	0.0D+00	0.0D+00	0.03
SIMBQP	2	B	g_p	(1, 5)	(2, 1)	0.0D+00	0.0D+00	0.01
		L	g_p	(1, 0)	(1, 2)	0.0D+00	0.0D+00	0.03

Table 4 (cont.): Academic bound–constrained quadratic problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
CHEBYQAD	50	B	g_p	(25, 1499)	(52, 25)	0.5D−02	0.4D−05	58.80
		L	g_p	(28, 453)	(28, 23)	0.5D−02	0.8D−05	19.20
HATFLDA	4	B	g_p	(5, 28)	(13, 5)	0.1D−10	0.4D−05	0.03
		L	g_p	(28, 73)	(28, 29)	0.3D−12	0.4D−6	0.17
HATFLDB	4	B	g_p	(5, 35)	(14, 5)	0.6D−02	0.9D−07	0.03
		L	g_p	(25, 55)	(25, 26)	0.6D−02	0.1D−05	0.14
HATFLDC	24	B	g_p	(6, 28)	(7, 6)	0.1D−10	0.1D−05	0.03
		L	g_p	(4, 18)	(4, 5)	0.2D−11	0.2D−05	0.10
HS1	2	B	g_p	(25, 52)	(35, 25)	0.3D−13	0.1D−06	0.02
		L	g_p	(35, 61)	(35, 31)	0.4D−11	0.2D−05	0.20
HS110	50	B	g_p	(1, 0)	(2, 1)	−0.1D+11	0.0D−00	0.02
		L	g_p	(1, 0)	(1, 2)	−0.1D+11	0.0D+0	0.04
HS2	2	B	g_p	(6, 8)	(7, 6)	0.5D+01	0.9D−07	0.02
		L	g_p	(11, 20)	(11, 10)	0.5D+01	0.2D−05	0.10
HS25	3	B	g_p	(0, 0)	(1, 0)	0.3D+02	0.2D−07	0.02
		L	g_p	(0, 0)	(0, 1)	0.3D+02	0.2D−07	0.03
LINVERSE	1999	B	g_p	(43, 2320)	(59, 43)	0.7D+03	0.9D−05	734.65
		L	g_p	(28, 2049)	(28, 23)	0.7D+03	0.3D−05	365.60
NONSCOMP	10000	B	g_p	(11, 51)	(12, 11)	0.9D−11	0.6D−05	32.51
		L	g_p	(8, 78)	(8, 9)	0.2D−11	0.5D−05	35.40
PSPDOC	4	B	g_p	(5, 16)	(9, 5)	0.2D+01	0.8D−10	0.02
		L	g_p	(8, 14)	(8, 9)	0.2D+01	0.9D−05	0.10
QR3DLS	610	B	g_p	(265, 78034)	(534, 265)	0.3D−07	0.3D−05	11269.31
		L	g_p	(323, 23614)	(323, 282)	0.2D−07	0.6D−05	3440.30
SPECAN	9	B	g_p	(13, 66)	(18, 13)	0.2D−12	0.3D−05	66.97
		L	g_p	(14, 68)	(14, 12)	0.4D−12	0.8D−05	110.40

Table 5: Academic bound–constrained least–squares problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
ALLINIT	4	B	g_p	(7, 40)	(18, 7)	0.2D+02	0.8D−06	0.08
		L	g_p	(11, 13)	(11, 10)	0.2D+02	0.8D−09	0.10
BDEXP	5000	B	g_p	(10, 10)	(11, 10)	0.2D−02	0.6D−05	13.79
		L	g_p	(10, 27)	(10, 11)	0.2D−02	0.6D−05	13.40
EG1	3	B	g_p	(6, 13)	(8, 6)	−0.1D+01	0.1D−04	0.02
		L	g_p	(6, 9)	(6, 7)	−0.1D+01	0.8D−05	0.06
EXPLIN	120	B	g_p	(14, 38)	(16, 14)	−0.8D+06	0.7D−05	0.21
		L	g_p	(13, 80)	(13, 14)	−0.8D+06	0.3D−05	0.20
EXPLIN2	120	B	g_p	(13, 34)	(14, 13)	−0.8D+06	0.4D−05	0.19
		L	g_p	(11, 52)	(11, 12)	−0.8D+06	0.5D−05	0.30
EXPQUAD	120	B	Δ	(21, 107)	(90, 21)	−0.4D+07	0.1D−03	0.90
		L	g_p	(18, 98)	(18, 16)	−0.4D+07	0.4D−06	0.60
HS38	4	B	g_p	(46, 247)	(78, 46)	0.7D−11	0.2D−05	0.13
		L	g_p	(54, 194)	(54, 47)	0.6D−17	0.6D−07	0.30
HS4	2	B	g_p	(1, 1)	(2, 1)	0.3D+01	0.0D+00	0.01
		L	g_p	(1, 0)	(1, 2)	0.3D+01	0.0D+00	0.01
HS45	5	B	g_p	(3, 3)	(4, 3)	0.1D+01	0.0D+00	0.02
		L	g_p	(8, 0)	(8, 9)	0.1D+01	0.0D+00	0.05
HS5	2	B	g_p	(5, 11)	(8, 5)	−0.2D+01	0.5D−08	0.02
		L	g_p	(5, 5)	(5, 6)	−0.2D+01	0.3D−06	0.03
MAXLIKA	8	B	g_p	(31, 210)	(42, 31)	0.1D+04	0.5D−05	11.31
		L	g_p	(8, 36)	(8, 9)	0.1D+04	0.6D−05	1.50
MCCORMCK	10000	B	g_p	(7, 14)	(8, 7)	−0.9D+04	0.7D−06	26.17
		L	g_p	(8, 16)	(8, 7)	−0.9D+04	0.4D−05	22.40
PROBPENL	500	B	g_p	(1, 0)	(2, 1)	0.4D−06	0.2D−06	0.22
		L	g_p	(1, 0)	(1, 2)	0.4D−06	0.3D−06	0.14
SINEALI	20	B	g_p	(7, 13)	(9, 7)	−0.2D+04	0.2D−05	0.05
		L	g_p	(17, 138)	(17, 13)	−0.2D+04	0.1D−04	0.20

Table 6: Other academic bound–constrained problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
PALMER1C	8	B#	g_p	(1, 20)	(2, 1)	0.1D+00	0.2D−06	0.02
		L	g_p	(85, 2006)	(85, 86)	0.1D+00	0.4D−05	2.20
PALMER1D	7	B	g_p	(1, 14)	(2, 1)	0.7D+00	0.3D−05	0.02
		L	g_p	(4, 40)	(4, 5)	0.7D+00	0.4D−05	0.10
PALMER2C	8	B	g_p	(1, 20)	(2, 1)	0.1D−01	0.3D−05	0.02
		L	g_p	(44, 1031)	(44, 45)	0.1D−01	0.7D−05	0.90
PALMER3C	8	B	g_p	(1, 19)	(2, 1)	0.2D−01	0.5D−07	0.01
		L	g_p	(8, 146)	(8, 9)	0.2D−01	0.6D−05	0.20
PALMER4C	8	B	g_p	(1, 17)	(2, 1)	0.5D−01	0.2D−05	0.01
		L	g_p	(33, 729)	(33, 34)	0.5D−01	0.6D−05	0.80
PALMER5C	6	B	g_p	(1, 6)	(2, 1)	0.2D+01	0.2D−12	0.01
		L	g_p	(4, 14)	(4, 5)	0.2D+01	0.5D−06	0.05
PALMER6C	8	B	g_p	(1, 18)	(2, 1)	0.2D−01	0.5D−08	0.01
		L	g_p	(48, 1099)	(48, 49)	0.2D−01	0.2D−05	1.00
PALMER7C	8	B	g_p	(1, 17)	(2, 1)	0.6D+00	0.9D−05	0.01
		L	g_p	(13, 259)	(13, 14)	0.6D+00	0.3D−05	0.20
PALMER8C	8	B	g_p	(1, 17)	(2, 1)	0.2D+00	0.3D−05	0.02
		L	g_p	(5, 68)	(5, 6)	0.2D+00	0.2D−06	0.10

Table 7: Real unconstrained quadratic problems

PROBLEM	N	S	RS	(IT$_{out}$, IT$_{inn}$)	(FE, GE)	$f(x)$	$\|g_p\|$	T
ODNAMUR	11130	B	g_p	(1, 37846)	(2, 1)	0.9D+04	0.9D−05	12530.26
		L	g_p	(9, 51556)	(9, 10)	0.9D+04	0.2D−05	10224.00

Table 8: Real bound–constrained quadratic problems

PROBLEM	N	S	RS	(IT$_{out}$, IT$_{inn}$)	(FE, GE)	$f(x)$	$\|g_p\|$	T
PALMER1	4	B	g_p	(19, 39)	(27, 19)	0.1D+05	0.2D−07	0.15
		L	g_p	(27, 40)	(27, 40)	0.1D+05	0.4D−08	0.30
PALMER1A	6	B	g_p	(104, 765)	(179, 104)	0.1D+00	0.5D−05	1.57
		L	g_p	(217, 901)	(217, 207)	0.1D+00	0.2D−05	2.90
PALMER1B	4	B	g_p	(27, 83)	(38, 27)	0.3D+01	0.9D−05	0.20
		L	g_p	(46, 75)	(46, 41)	0.3D+01	0.1D−06	0.40
PALMER1E	8	B	g_p	(179, 1941)	(268, 179)	0.8D−03	0.1D−05	3.77
		L	g_p	(274, 4310)	(274, 266)	0.8D−03	0.4D−05	7.30
PALMER2	4	B#	g_p	(17, 45)	(30, 17)	0.4D+04	0.2D−05	0.13
		L	g_p	(29, 39)	(29, 24)	0.4D+04	0.2D−06	0.30
PALMER2A	6	B	g_p	(59, 348)	(91, 59)	0.2D−01	0.5D−05	0.59
		L	g_p	(99, 474)	(99, 83)	0.2D−01	0.3D−05	1.10
PALMER2B	4	B	g_p	(17, 78)	(27, 17)	0.6D+00	0.2D−05	0.11
		L	g_p	(44, 76)	(44, 38)	0.6D+00	0.3D−05	0.40
PALMER2E	8	B	g_p	(455, 3627)	(560, 455)	0.2D−03	0.9D−05	5.42
		L	g_p	(139, 1660)	(139, 130)	0.2D−03	0.6D−05	2.70
PALMER3	4	B	g_p	(14, 39)	(22, 14)	0.2D+04	0.9D−05	0.10
		L	g_p	(28, 26)	(28, 26)	0.2D+04	0.7D−05	0.20
PALMER3A	6	B	g_p	(93, 741)	(160, 93)	0.2D−01	0.3D−05	1.00
		L	g_p	(165, 800)	(165, 150)	0.2D−01	0.3D−05	1.80
PALMER3B	4	B	g_p	(17, 71)	(27, 17)	0.4D+01	0.5D−05	0.17
		L	g_p	(29, 63)	(29, 26)	0.4D+01	0.5D−07	0.20
PALMER3E	8	B	g_p	(442, 4040)	(632, 442)	0.5D−04	0.3D−05	5.72
		L	g_p	(112, 1266)	(112, 107)	0.5D−04	0.6D−05	2.30
PALMER4	4	B	g_p	(14, 36)	(22, 14)	0.2D+04	0.9D−05	0.13
		L	g_p	(21, 21)	(21, 19)	0.2D+04	0.1D−04	0.30
PALMER4A	6	B	g_p	(41, 230)	(61, 41)	0.4D−01	0.5D−05	0.37
		L	g_p	(77, 347)	(77, 68)	0.4D−01	0.2D−05	0.80
PALMER4B	4	B	g_p	(16, 73)	(25, 16)	0.6D+01	0.7D−05	0.14
		L	g_p	(22, 51)	(22, 20)	0.6D+01	0.5D−07	0.20
PALMER4E	8	B	g_p	(227, 2228)	(335, 227)	0.1D−03	0.8D−05	3.06
		L	g_p	(91, 1130)	(91, 83)	0.1D−03	0.4D−05	1.80
PALMER5A	8	B	∞_F	(744, 5219)	(1000, 744)	0.1D+00	0.5D−03	5.00
		L	∞_I	(999, 11559)	(999, 969)	0.1D+00	0.98D−01	13.70
PALMER5B	9	B	g_p	(651, 4480)	(789, 651)	0.1D−01	0.5D−05	4.35
		L	g_p	(296, 2617)	(296, 266)	0.1D−01	0.6D−05	3.70
PALMER5D	8	B	g_p	(5, 23)	(6, 5)	0.9D+02	0.7D−12	0.04
		L	g_p	(2, 7)	(2, 3)	0.9D+02	0.7D−08	0.03
PALMER5E	8	B	∞_F	(512, 5276)	(1003, 512)	0.3D−01	0.2D+01	4.99
		L	∞_I	(999, 7839)	(999, 870)	0.3D−01	0.1D−03	11.40
PALMER6A	6	B	g_p	(178, 1501)	(298, 178)	0.6D−01	0.7D−05	1.27
		L	g_p	(164, 765)	(164, 144)	0.6D−01	0.6D−05	1.40
PALMER6E	8	B	g_p	(58, 807)	(106, 58)	0.2D−03	0.9D−05	0.69
		L	g_p	(103, 946)	(103, 95)	0.2D−03	0.3D−05	1.30

Table 9: Real bound–constrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
PALMER7A	6	B#	∞_F	(736, 4080)	(1001, 736)	0.1D+02	0.2D+01	4.24
		L	∞_I	(999, 5441)	(999, 883)	0.1D+02	0.5D+01	9.40
PALMER7E	8	B	g_p	(14, 137)	(22, 14)	0.1D+02	0.7D−05	0.15
		L	g_p	(403, 2358)	(403, 351)	0.1D+02	0.1D−04	4.40
PALMER8A	6	B#	g_p	(37, 152)	(41, 37)	0.7D−01	0.8D−05	0.13
		L	g_p	(62, 309)	(62, 54)	0.07D−01	0.9D−06	0.80
PALMER8E	8	B	g_p	(41, 419)	(66, 41)	0.6D−02	0.5D−06	0.35
		L	g_p	(35, 391)	(35, 33)	0.6D−02	0.5D−05	0.60
WEEDS	3	B	g_p	(35, 70)	(43, 35)	0.3D+01	0.2D−07	0.14
		L	g_p	(27, 61)	(27, 24)	0.3D+01	0.7D−05	0.20

Table 9 (cont.): Real bound–constrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
TOINTQOR	50	B	g_p	(1,27)	(2,1)	0.1D+04	0.5D−05	0.06
		L	g_p	(4, 35)	(4, 5)	0.1D+04	0.3D−05	0.10

Table 10: Modelling unconstrained quadratic problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
GULF	3	B	g_p	(30,82)	(51,30)	0.9D−08	0.2D−06	2.22
		L	g_p	(42, 90)	(42, 38)	0.2D−07	0.2D−05	1.30
HEART6LS	6	B#	∞_F	(498,2875)	(1000,498)	0.3D−01	0.2D+03	4.47
		L	∞_I	(1000, 2179)	(1000, 874)	0.9D−01	0.1D+02	7.50
KOWOSB	4	B	g_p	(10,58)	(20,10)	0.3D−03	0.4D−05	0.09
		L	g_p	(10, 26)	(10, 11)	0.3D−03	0.6D−05	0.10
MOREBV	5000	B	g_p	(2,88)	(3,2)	0.2D−09	0.2D−5	21.60
		L	g_p	(4, 73)	(4, 5)	0.3D−09	0.4D−05	15.30
OSBORNEB	11	B	g_p	(23,332)	(47,23)	0.4D−01	0.5D−07	3.59
		L	g_p	(31,204)	(31,28)	0.4D−01	0.8D−05	2.00
VIBRBEAM	8	B#	∞_F	(943,4173)	(1000,943)	0.2D+00	0.1D+00	42.52
		L	g_p	(169,1082)	(169,157)	0.2D+00	0.7D−05	9.60
YFITU	3	B	g_p	(74,213)	(117,74)	0.2D−07	0.8D−05	0.44
		L	g_p	(85,224)	(85,70)	0.3D−08	0.1D−05	0.70

Table 11: Modelling unconstrained least–squares problems

PROBLEM	N	S	RS	$(\mathrm{IT}_{out}, \mathrm{IT}_{inn})$	(FE, GE)	$f(x)$	$\|g_p\|$	T
FMINSURF	15625	B	g_p	(80,5475)	(170,80)	0.1D+01	0.1D−05	8332.79
		L	g_p	(974, 2396)	(974,893)	0.1D+01	0.2D−05	6264.70
TOINTGOR	50	B	g_p	(8,124)	(9,8)	0.1D+04	0.5D−05	0.30
		L	g_p	(6, 128)	(6, 7)	0.1D+04	0.2D−05	0.20

Table 12: Other modelling unconstrained problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
TORSION1	14884	B	g_p	(1,803)	(2,1)	−0.4D+00	0.7D−05	1251.14
		L	g_p	(37, 1347)	(37, 38)	−0.4D+00	0.4D−06	1050.90
TORSION2	14884	B	g_p	(1,765)	(2,1)	−0.4D+00	0.9D−05	1276.25
		L	g_p	(10, 5053)	(10, 11)	−0.4D+00	0.2D−05	4796.60
TORSION3	14884	B	g_p	(1,270)	(2,1)	−0.1D+01	0.8D−05	288.87
		L	g_p	(19, 390)	(19, 20)	−0.1D+01	0.9D−06	261.30
TORSION4	14884	B	g_p	(1,225)	(2,1)	−0.1D+01	0.9D−05	266.25
		L	g_p	(15, 5954)	(15, 16)	−0.1D+01	0.5D−06	5289.10
TORSION5	14884	B	g_p	(1,84)	(2,1)	−0.2D+01	0.8D−05	85.25
		L	g_p	(9,114)	(9, 10)	−0.3D+01	0.2D−05	75.30
TORSION6	14884	B	g_p	(1,78)	(2,1)	−0.3D+01	0.8D−05	70.77
		L	g_p	(11,7355)	(11,12)	−0.3D+01	0.6D−06	6249.30
TORSIONA	14884	B	g_p	(1,858)	(2,1)	−0.4D+00	0.8D−05	1315.55
		L	g_p	(37,1339)	(37,38)	−0.4D+00	0.5D−06	1109.80
TORSIONB	14884	B	g_p	(1,588)	(2,1)	−0.4D+00	0.9D−05	995.13
		L	g_p	(8,5000)	(8,9)	−0.4D+00	0.3D−05	4873.40
TORSIONC	14884	B	g_p	(1,273)	(2,1)	−0.1D+01	0.9D−05	301.21
		L	g_p	(19,390)	(19, 20)	−0.1D+01	0.1D−05	277.80
TORSIOND	14884	B	g_p	(1,227)	(2,1)	−0.1D+01	0.9D−05	322.86
		L	g_p	(9,9430)	(9, 10)	−0.1D+01	0.3D−05	8345.90
TORSIONE	14884	B	g_p	(1,82)	(2,1)	−0.3D+01	0.8D−05	70.89
		L	g_p	(9,114)	(9, 10)	−0.3D+01	0.2D−05	77.50
TORSIONF	14884	B	g_p	(1,90)	(2,1)	−0.3D+01	0.8D−05	100.66
		L	g_p	(10,5343)	(10, 11)	−0.3D+01	0.7D−06	4120.80

Table 13: Modelling bound–constrained quadratic problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
YFIT	3	B	g_p	(74,213)	(117,74)	0.2D−07	0.8D−05	0.43
		L	g_p	(85,224)	(85,70)	0.3D−08	0.1D−05	0.70

Table 14: Modelling bound–constrained least–squares problems

PROBLEM	N	S	RS	(IT_{out}, IT_{inn})	(FE, GE)	$f(x)$	$\|g_p\|$	T
QRTQUAD	120	B	g_p	(30,275)	(55,30)	−0.4D+07	0.6D−05	1.65
		L	g_p	(178, 570)	(178, 145)	−0.4D+07	0.4D−05	6.00
S368	100	B	g_p	(9,34)	(12,9)	−0.1D+03	0.7D−05	48.36
		L	g_p	(8, 20)	(8, 7)	−0.1D+03	0.2D−07	21.70

Table 15: Other modelling bound–constrained problems

The diagrams of Figure 1 illustrate the proportion of problems of different dimensions in the set of tests. For problems with more than one dimension, we selected the largest possibility that matched with the available requirements of memory. This justifies the small proportion of medium size problems in our set of tests. For the quadratic problems, the class $(100, 500]$ is empty. Most of the problems are concentrated either in the first class, with $1 \leq n \leq 50$ or in the last class ($10000 < n \leq 15625$). Considering the non-quadratic problems, the majority is constituted of the ones with smallest dimensions: 59% have dimension less than or equal to 50. The second largest set is the one obtained by combining classes E, F and G, that is, for which $n > 1000$, corresponding to 25% of the non-quadratic problems.

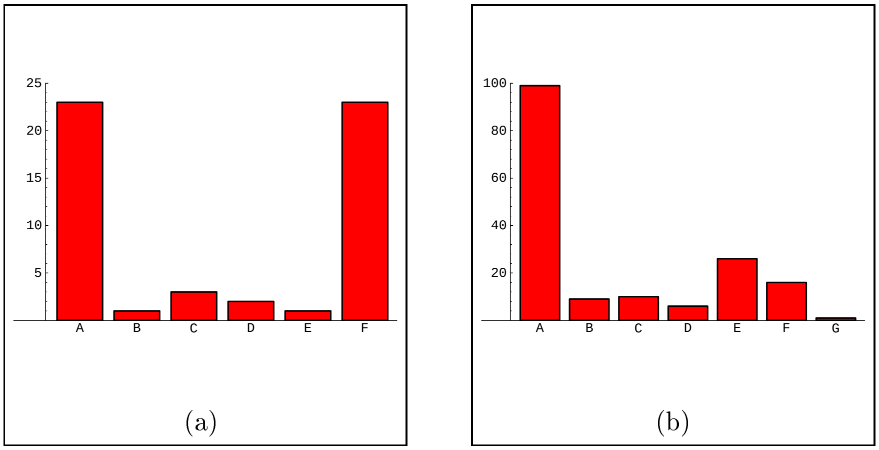

Fig. 1: Dimension of problems in the set of tests. Chart (a): Quadratic problems - A = (0,50]; B=(50,100]; C=(500,1000]; D=(1000,5000]; E=(5000,10000]; F=(10000;15625]. Chart (b): Non-quadratic problems - A = (0,50]; B=(50,100]; C=(100,500]; D=(500,1000]; E=(1000,5000]; F=(5000,10000]; G=(10000;15625].

		$GMIT_{out}$	$GMIT_{inn}$	GMFE	GMGE	GMT
AUQ	B	1.000	18.61	2.000	1.000	0.493
(7)	L	3.074	23.64	3.074	4.092	1.127
		67%	21%	35%	76%	56%
AUL	B	23.72	117.4	36.97	23.72	1.370
(50)	L	25.91	83.01	25.91	24.32	2.145
		8%	−41%	−43%	3%	36%
OAU	B	13.74	67.07	20.04	13.74	3.693
(45)	L	16.40	62.91	16.40	15.93	5.683
		16%	−7%	−22%	14%	35%
ABQ	B	1.000	84.87	2.000	1.000	7.073
(23)	L	7.860	153.5	8.100	9.851	19.12
		87%	45%	75%	90%	63%
ABL	B	9.418	61.14	15.02	9.418	0.832
(13)	L	12.79	60.44	12.79	13.11	1.622
		26%	−1%	−17%	28%	49%
OAB	B	7.469	17.35	11.76	7.469	0.232
(14)	L	7.945	16.37	7.945	8.819	0.272
		6%	−6%	−48%	15%	15%
RUQ	B	1.000	15.65	2.000	1.000	.0136
(9)	L	15.42	237.3	15.42	17.05	0.316
		94%	93%	87%	94%	96%
RBQ	B	1.000	37846.	2.000	1.000	12530.
(1)	L	9.000	51556.	9.000	10.00	10224.
		89%	27%	78%	90%	−23%
RBL	B	49.06	267.6	73.00	49.06	0.437
(24)	L	62.93	252.1	62.93	59.17	0.730
		22%	−6%	−16%	17%	42%

Table 16: Average computational effort of each set of problems

		GMIT$_{out}$	GMIT$_{inn}$	GMFE	GMGE	GMT
MUQ (1)	B	1.000	27.00	2.000	1.000	0.060
	L	4.000	35.00	4.000	5.000	0.100
		75%	23%	50%	80%	40%
MUL (5)	B	15.92	124.2	27.87	15.92	1.468
	L	21.34	95.17	21.34	21.01	1.227
		25%	−31%	−31%	24%	−20%
OMU (2)	B	25.30	824.0	39.12	25.30	50.00
	L	76.45	553.8	76.45	79.06	35.40
		67%	−49%	49%	68%	−41%
MBQ (12)	B	1.000	248.9	2.000	1.000	306.0
	L	13.80	1554.	13.80	14.91	1239.
		93%	84%	86%	93%	75%
MBL (1)	B	74.00	213.0	117.0	74.00	0.430
	L	85.00	224.0	85.00	70.00	0.700
		13%	5%	−38%	−6%	39%
OMB (2)	B	16.43	96.70	25.69	16.43	8.933
	L	37.74	106.8	37.74	31.86	11.41
		56%	9%	32%	48%	22%

Table 16 (cont.): Average computational effort of each set of problems

		GMIT$_{out}$	GMIT$_{inn}$	GMFE	GMGE	GMT
QUADRATIC (53)	B	1.000	73.02	2.000	1.000	4.249
	L	8.753	236.7	8.868	10.44	17.19
		89%	69%	77%	90%	75%
NON-QUADRATIC (156)	B	18.74	93.04	28.64	18.74	1.336
	L	22.52	79.52	22.52	21.85	2.028
		17%	−17%	−27%	14%	34%
UNCONSTRAINED (119)	B	12.08	74.89	19.12	12.08	1.374
	L	18.38	77.48	18.38	18.18	2.578
		34%	3%	−4%	34%	47%
BOUND-CONSTRAINED (90)	B	5.959	107.5	10.19	5.959	2.542
	L	16.88	156.5	17.01	18.05	5.198
		65%	31%	40%	67%	51%
TOTAL (209)	B	8.912	87.50	14.58	8.912	1.791
	L	17.72	104.9	17.78	18.12	3.487
		50%	17%	18%	51%	49%

Table 17: Summarized average computational effort of the numerical experiments

Tables 16 and 17 summarize the average computational effort of algorithms `BOX-QUACAN` and `LANCELOT`. The notation used is similar to the previous tables: GMIT$_{out}$ and GMIT$_{inn}$ contain the geometric means of the number of outer and inner iterations of each algorithm, GMFE and GMGE inform the geometric means of the number of functional and gradient evaluations and GMT gives the geometric mean of the time spent by each test. For not having to exclude the problems with zero for any of the values IT$_{out}$, IT$_{inn}$, FE or GE from the average results, we replaced all the zeros by ones in the computation of the geometric means. It is worthwhile stressing that the logarithm with base 10 of the geometric means presented in Tables 16 and 17 define the average order of magnitude of the computed values. An additional row was included that, column by column, compares the performance of both algorithms. Keeping the effort spent by `LANCELOT` as a reference, we computed the relative difference between `LANCELOT` and `BOX-QUACAN`. The positive values indicate a superior performance of `BOX-QUACAN`. The data from problems for which convergence failed (11 out of 220, i.e. 5%) were excluded from Tables 16 and 17, and the numbers in parenthesis indicate the amount of problems in that particular class. Convergence failed (∞_F or ∞_I) for both algorithms in 4.1% of the tests (9 out of 220). `BOX-QUACAN` failed alone in 0.9% of the tests (2 out of 220), while `LANCELOT` never failed for the problems successfully solved by `BOX-QUACAN`. The two failures occurred for problems EXTROSNB (Table 2) and VIBRBEAM (Table 11). In both cases we observed that the objective function value at the final point is either better or the same as the one achieved by `SBMIN`, but the norm of the projected gradient is still greater than the required precision $\varepsilon_g = 10^{-5}$, probably because the objective function is badly scaled.

In Table 16, the more relevant columns for the comparison are GMIT$_{inn}$ and GMGE. The former represents the computational effort of the quadratic solver and the latter registers the number of distinct points (i.e. accepted by the sufficient decrease condition) in the sequence generated by each algorithm. Declaring the cases where the relative measure belongs to the interval (-10%,10%) as a tie between the two algorithms, the analysis of the results of Table 16 is given as follows. Horizontally, except for the sets AUL and MUL, `BOX-QUACAN` performed better that `LANCELOT`. Vertically, concerning inner iterations, the algorithm `BOX-QUACAN` had a worse performance than `LANCELOT` in the sets AUL, RBQ, MUL and OMU. For the sets AUQ, ABQ, RUQ, MUQ and MBQ the opposite took place and for the remaining sets (OAU, ABL, OAB, RBL, MBL and OMB) ties occurred. The column GMGE indicates an advantage of `BOX-QUACAN` over `LANCELOT` as far as derivative evaluations are concerned for all sets except AUL and MBL, for which ties were obtained. In terms of column GMFE, we observed that `BOX-QUACAN` uses a larger number of functional evaluations than `LANCELOT` for the sets AUL, OAU, ABL, OAB, RBL, MUL and MBL, indicating that `BOX-QUACAN` rejected a larger amount of points in the sufficient decrease test, probably due to the simplicity of its scheme for updating the trust–region radius. For the remaining sets (AUQ, ABQ, RUQ, RBQ, MUQ, OMU, MBQ and OMB) `BOX-QUACAN` needed fewer functional evaluations than `LANCELOT`.

For most of the small problems, the running time was included only for completion, since the measure of time can be rather innacurate in this case. As far as the large-scale sparse problems are concerned, it is frequently seen that `BOX-QUACAN` works less, performing fewer inner and/or outer iterations than `SBMIN` but spends a larger amount of running time. The reason of such behavior is that, despite containing important information about the performance of each algorithm separately, the columns T and GMT of Tables 1-15 and 16-17 do not provide fair comparative measures. Since the test problems were selected from the `CUTE` collection, the routines for computing the objective function value, its gradient and the product of its Hessian by a vector are provided by this environment. Unfortunately, it is a drawback of CUTE that, except for LANCELOT, it is not given to any other software the option of exploiting the sparsity of a vector when it is multiplied by the Hessian matrix. Because of this, the difference in spent time increases significatively with the dimension of the problems. For example, in the set MBQ (Table 13), with 12 problems having dimension $n = 14884$, the performance of BOX-QUACAN is by far superior than the one of LANCELOT for all the considered quantifiers. Nevertheless, the relative measure of GMT is proportionally inferior when compared to the other measures.

Analysing Table 17 by rows we observe that on the whole BOX-QUACAN surpasses LANCELOT. For quadratic problems, for the bound–constrained set and for the total set of solved problems all the quantifiers are more favorable to BOX-QUACAN. For non-quadratic problems LANCELOT performed better in terms of inner iterations and functional evaluations. For unconstrained problems, there was a tie between the two algorithms for the values GMIT_{inn} and GMFE.

There were two cases of stopping because the trust–region radius became too small: problems PENALTY3 (Table 3) and EXPQUAD (Table 6). For the former both BOX- QUACAN and LANCELOT stopped because $\Delta_k \leq \varepsilon_\Delta$, while for the latter only BOX-QUACAN stopped with this criterion. We observed, however, that for these problems the norm of the projected gradient at the final point is small, but not small enough to achieve convergence with $\|g_p\| \leq \varepsilon_g$ as demanded. In other words, the final point is almost optimal and the expected convergence was not reached probably due to influence of objective function scaling in both problems. Such influence was also present in problems marked with # in Tables 1-15 (13 out of 220, i.e. 6%), for which the uniform choice of $M_k = 10^5$ for the upper bound was not large enough and so it had to be increased to $M_k = 10^{10}$.

An interesting aspect of our quadratic solver was detected through the real bound-constrained quadratic problem ODNAMUR (Table 8). To uniformize the choices for the whole set of tests, we defined very loosely the parameter of the quadratic solver in charge of deciding to abandon the current face. For problem ODNAMUR, however, such choice showed to be rather poor. Probably due to 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. The results presented in Table 8 correspond to the more conservative choice for such parameter and this is the only exception among the test problems.

Figures 2-6 provide visual alternatives for Table 17. The problems were classified in quadratic & non-quadratic and unconstrained & bound-constrained so that the results of each algorithm become more evident. The first two diagrams illustrate the consequences of the different approaches adopted by BOX-QUACAN and LANCELOT concerning the solution of the quadratic subproblems. More specifically, Figure 2 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. Figure 3 points out a slight advantage of BOX-QUACAN over LANCELOT as far as non-quadratic problems are concerned. From

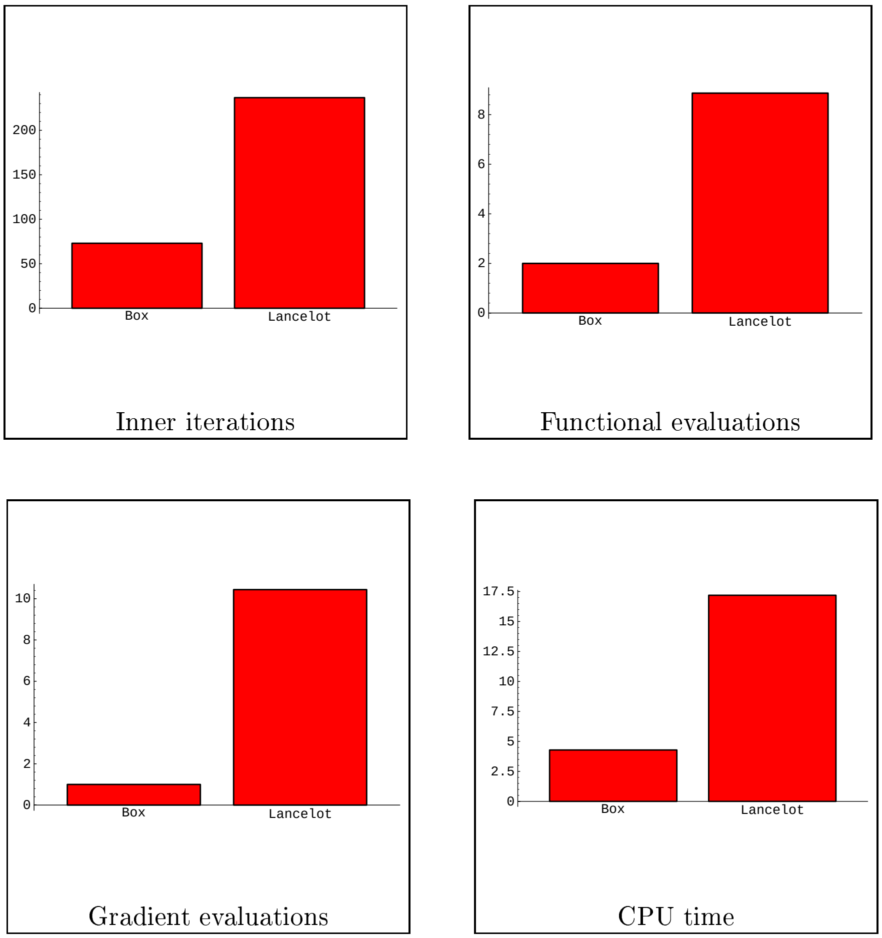

Fig. 2: Computational effort of quadratic problems

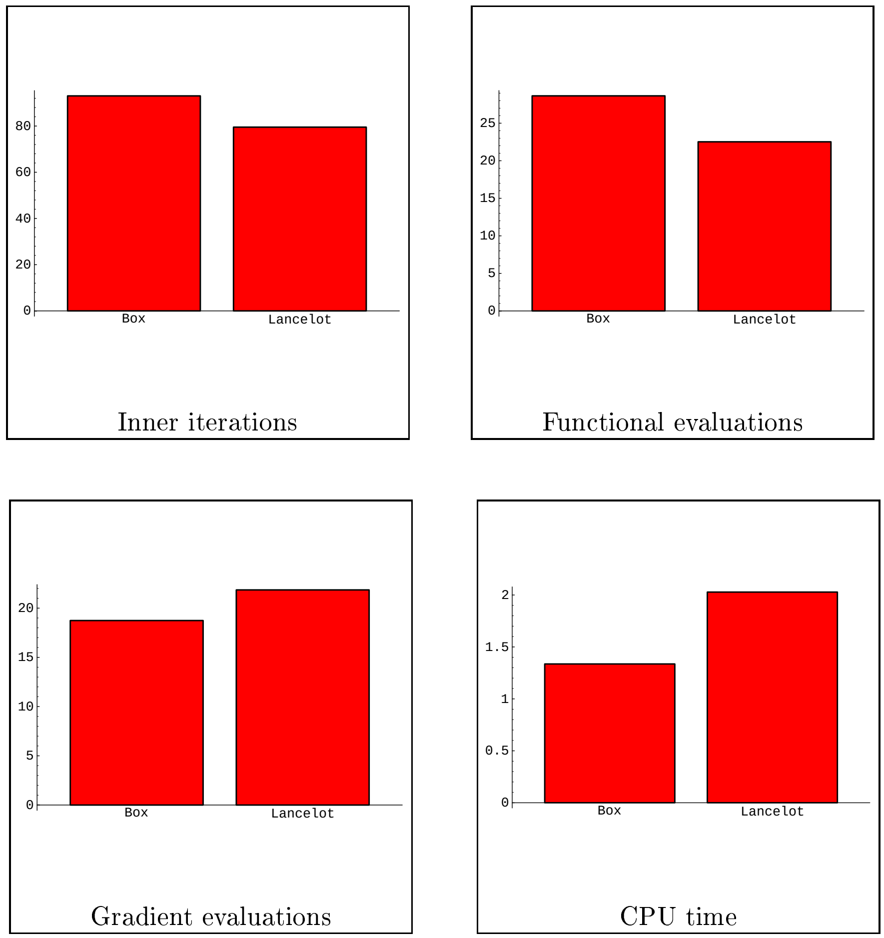

Fig. 3: Computational effort of non-quadratic problems

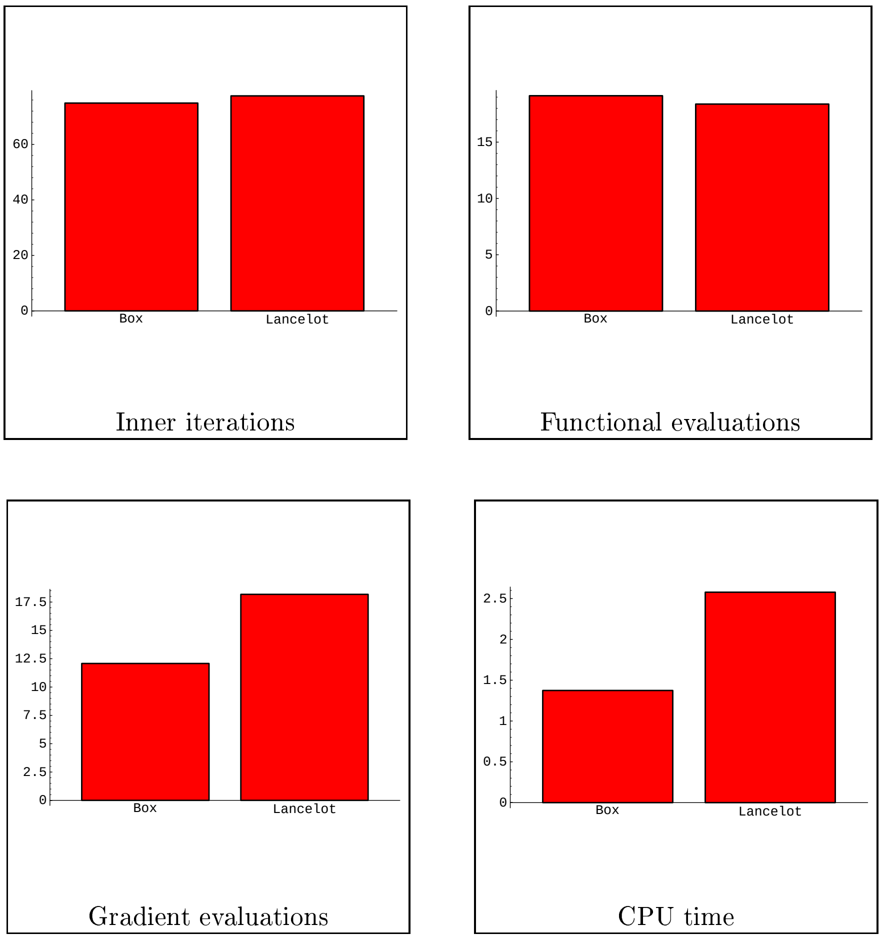

Fig. 4: Computational effort of unconstrained problems

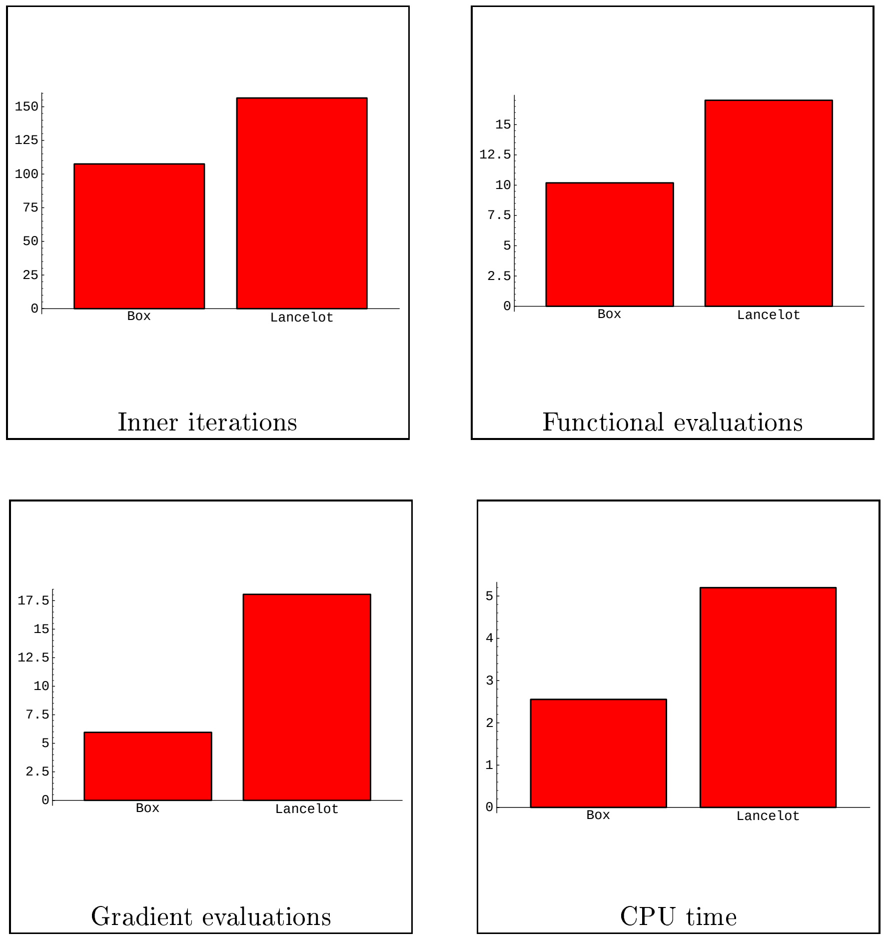

Fig. 5: Computational effort of bound-constrained problems

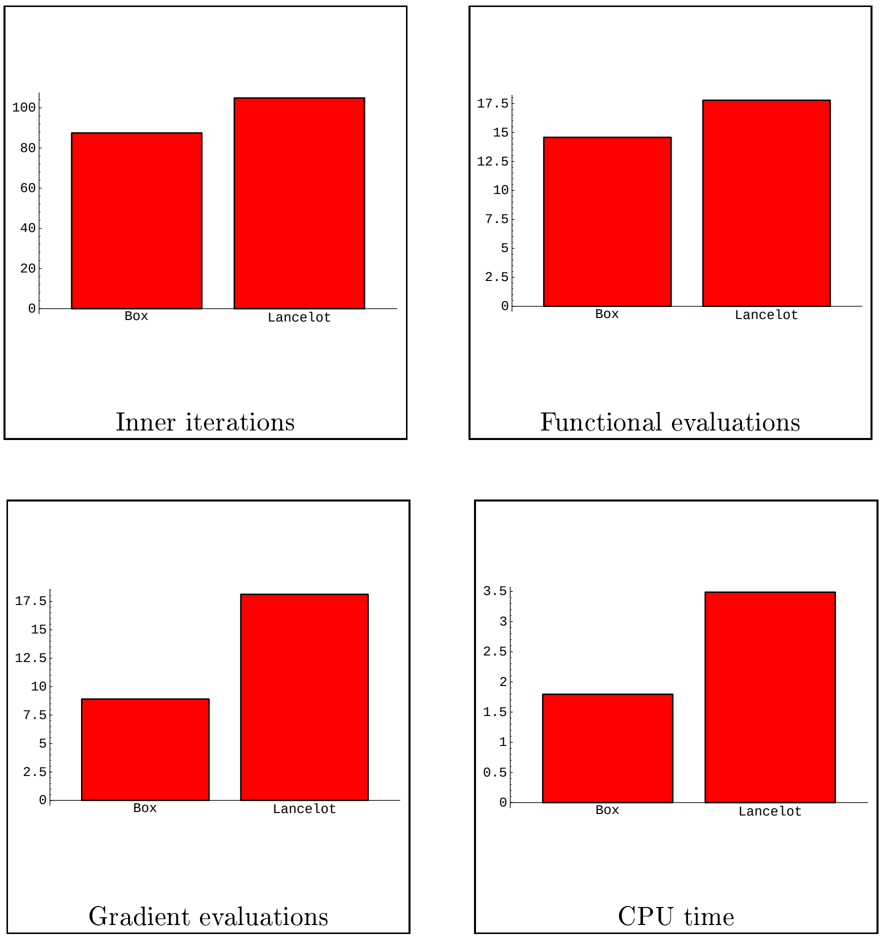

Fig. 6: Computational effort of the whole set of solved problems

these results we can observe that if the problem is non-quadratic, the effort of solving the quadratic subproblem on its whole feasible set does not seem worthwhile, specially in the initial outer iterations, when the agreement between the quadratic model and the problem is still to be tuned. In this sense, we believe that in the begining an even more relaxed tolerance for approximately solving the quadratic subproblem should be adopted in `BOX-QUACAN`. Through Figures 4 and 5 we can constrast the performance of the two algorithms for problems with either artificial or natural bounds. In both cases `BOX-QUACAN` had a superior behavior, although for the bound-constrained set the diference is qualitatively larger. Figure 6 summarizes the average computational effort of the numerical experiments considering all the problems as a single block, showing that both approaches are competitive.

6 Final Remarks

As expected, the algorithm `BOX-QUACAN` had a very good performance for the quadratic problems. This shows the value of investing in the quadratic solver whenever it is known that the quadratic model represents well the objective function, in which case `BOX-QUACAN` is preferable to `SBMIN`.

The numerical tests detected that there is certainly room for improvement in the strategy for updating the trust–region radius of `BOX-QUACAN`. It is an interesting aspect to be investigated whether the implementation of `SBMIN` strategy for updating Δ would improve `BOX-QUACAN` performance or not. Distinct ideas could be also tested and a recent worthwhile mentioning work in this subject is [14]. Another aspect that is currently under investigation is the initialization of the quadratic subproblem (6) (cf. [13]). Future research also includes an extension of `BOX-QUACAN` for dealing with nonlinear constraints. A step in this direction was done in [8], where an algorithm for minimizing convex quadratics with simple bounds and equality constraints based on augmented Lagrangians with adaptive precision control is proposed.

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] I. Bongartz, A. R. Conn, N. I. M. Gould & Ph. L. Toint, *CUTE: a Collection of Constrained and Unconstrained Test Examples for Nonlinear Programming*. Technical Report, Department of Mathematics, FUNDP, Namur, Belgium, 1992. See also CUTE documentation.

[2] T. F. Coleman & Y. Li, "An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds". *SIAM Journal on Optimization*, **6**, pp. 418–445, 1996.

[3] 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.

[4] 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.

[5] A. R. Conn, N. I. M. Gould & Ph. L. Toint, "A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds". *SIAM Journal on Numerical Analysis*, **28**, pp. 545–572, 1991.

[6] 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.

[7] J. E. Dennis & L. Vicente, "Trust-region Interior-point Algorithms for Minimization Problems with Simple Bounds". *Applied Mathematics and Parallel Computing*, Festschrift for Klaus Ritter, edited by H. Fisher, B. Riedmüller and S. Schäffler, Physica–Verlag, Springer–Verlag, pp. 97–107, 1996.

[8] Z. Dostál, A. Friedlander & S. A. Santos, *Augmented Lagrangians with adaptive precision control for quadratic with simple bounds and equality constraints.* Relatório de Pesquisa **RP74-96**, Instituto de Matemática, Universidade Estadual de Campinas, 1996.

[9] 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.

[10] 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.

[11] 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.

[12] M. A. Gomes–Ruggiero & P. G. Kakuzo, *Análise do conjunto de problemas CUTE em testes de software em otimização.* Relatório de Pesquisa **RP07-96**, Instituto de Matemática, Universidade Estadual de Campinas, 1996.

[13] M. A. Gomes–Ruggiero & S. A. Santos, "Hot Start for the Quadratic Subproblem of BOX-QUACAN". In: *Fifth SIAM Conference on Optimization*, Victoria, British Columbia, Canada, May, 1996.

[14] A. Sartenaer, "Automatic determination of an initial trust region in nonlinear programming", to appear in *SIAM Journal on Scientific Computing*, 1996.