COMPUTATIONAL & APPLIED MATHEMATICS

Volume 30, N. 1, pp. 171–196, 2011

ISSN 0101-8205
www.scielo.br/cam

Active-set strategy in Powell's method for optimization without derivatives

MA. BELÉN AROUXÉT[1], NÉLIDA ECHEBEST[1] and ELVIO A. PILOTTA[2]

[1]Departamento de Matemática, Facultad de Ciencias Exactas, Universidad Nacional de La Plata 50 y 115, La Plata (1900), Buenos Aires, Argentina

[2]Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, CIEM (CONICET), Medina Allende s/n, Ciudad Universitaria (5000) Córdoba, Argentina

E-mails: belen|opti@mate.unlp.edu.ar / pilotta@famaf.unc.edu.ar

Abstract. In this article we present an algorithm for solving bound constrained optimization problems without derivatives based on Powell's method [38] for derivative-free optimization. First we consider the unconstrained optimization problem. At each iteration a quadratic interpolation model of the objective function is constructed around the current iterate and this model is minimized to obtain a new trial point. The whole process is embedded within a trust-region framework. Our algorithm uses infinity norm instead of the Euclidean norm and we solve a box constrained quadratic subproblem using an active-set strategy to explore faces of the box. Therefore, a bound constrained optimization algorithm is easily extended. We compare our implementation with NEWUOA and BOBYQA, Powell's algorithms for unconstrained and bound constrained derivative free optimization respectively. Numerical experiments show that, in general, our algorithm require less functional evaluations than Powell's algorithms.

Mathematical subject classification: Primary: 06B10; Secondary: 06D05.

Key words: derivative-free optimization, active-set method, spectral gradient method.

1 Introduction

We consider the bound constrained optimization problem where the derivatives of the objective function f are not available and the functional values $f(x)$ are

#CAM-321/10. Received: 15/VIII/10. Accepted: 05/I/11.
Work sponsored by CONICET and SECYT-UNC.

typically very expensive or difficult to compute. That is

$$\min f(x) \text{ subject to } x \in \Omega,$$

where

$$\Omega = \{x \in \mathbb{R}^n \mid L \leq x \leq U\},$$

$L < U$, and we assume that $\nabla f(x)$ cannot be computed for any x. First of all we consider the unconstrained problem, i.e., $\Omega = \mathbb{R}^n$.

This situation frequently occurs in problems where functional values $f(x)$ either come from physical, chemical or geophysical measurements or are the results from very complex computer simulations. The diversity of applications includes different problems such as rotor blade design [10], wing platform design [4], aeroacustic shape design [26], hydrodynamic design [17] and also problems of molecular geometry [1, 27], groundwater community [19, 30], medical image registration [33] and dynamic pricing [24].

There are several essentially different methods for solving this kind of problems [14]. A first group of methods includes the direct search or pattern search methods. They are based on the exploration of the variables space using function evaluations in sample points given by a predefined geometric pattern. That is the case of methods where sampling is guided by a suitable set of directions [15, 41] and based on simplices and operations over them, such as the Nelder-Mead algorithm [31]. They do not exploit the possible smoothness of the objective function and, therefore, require a very large number of function evaluations. They can also be useful for non-smooth problems. A comprehensive survey of these methods can be found in [23].

A second group of methods is based on modelling the objective function by multivariate interpolation in combination with trust-region techniques. These methods were introduced by Winfield [42, 43]. A polynomial model is built in order to interpolate the objective function at the points where the functional values are known. The model is then minimized over the trust-region and a new point is computed. The objective function is evaluated at this new point and thus possibly enlarging the interpolation set. This new computed point is checked as to whether the objective function is improved and the whole process is repeated until convergence is achieved. Thus, the geometry of the interpola-

tion set points and the model minimization are the keys to a good performance of the algorithms.

At the present time, there are several implementations of algorithms based on interpolation approaches, although the most tested and well established are DFO developed by Conn, Scheinberg and Toint [11, 12, 13] and NEWUOA developed by Powell [34, 35, 36, 37, 38, 39, 40]. See also the Wedge method developed by Marazzi and Nocedal [25] and the code developed by Berghen and Bersini [5], named CONDOR, which includes a parallel version based on NEWUOA.

In this article we consider the model-based trust-region method NEWUOA [38] because this code performed very well in recent comparison benchmark articles by [18, 29, 32]. Moreover, Moré and Wild [29] report that NEWUOA is the most effective derivative-free unconstrained optimization method for smooth functions. These results and the recent developments by Powell [39] have encouraged us for further development of model-based methods. Recently, Powell introduced the algorithm BOBYQA [40], a new version of NEWUOA, that successfully solves bound constrained optimization problems.

In NEWUOA [38] and BOBYQA [40] the trust-region subproblem is defined using the Euclidean norm and it is solved by a (truncated) conjugate gradient method. In our research, we make use of the infinity norm instead of the Euclidean norm, and use an active-set strategy in combination with the spectral projected gradient method in the same way that was proposed by Birgin and Martínez [7]. This strategy is not computationally expensive and it has been successful for medium-scale problems, so we consider it could be useful for our algorithm.

The numerical results and the observations made in this paper are based on experiments involving all the smooth problems suggested in [29]. We have also tested the algorithms for a set of medium-scale problems (100 variables). We compare our implementation with NEWUOA (for unconstrained optimization problems) and BOBYQA (for bound constrained optimization problems).

This article is organized as follows. In Section 2 we describe the main ideas of the interpolation-based methods for derivative-free optimization. In Section 3 we give a short description of the NEWUOA solver for derivative-free optimization. In Section 4 we describe the active-set strategy for solving the quadratic

minimization trust-region subproblem. In Section 5 we show numerical results of our implementation for unconstrained and bound constrained optimization problems and we give some comments about the performance. Finally, conclusions are given in Section 6.

2 Main ideas of the interpolation-based methods for derivative-free optimization

Trust-region strategies have been considered in derivative-free optimization in many articles [11, 12, 13, 37, 38, 39]. Basically, the main steps of the trust-region method for nonlinear programming are the following:

Step 1: Building interpolation step. Given a current iterate x_k, build a good local approximation model (e.g., based on a second order Taylor approximation):

$$m_k(x_k + s) = d_k + s^T g_k + \frac{1}{2} s^T G_k s,$$

where $d_k \in \mathbb{R}$, $g_k \in \mathbb{R}^n$ and $G \in \mathbb{R}^{n \times n}$ is a symmetric matrix, whose coefficients are determined by using the interpolation conditions.

Step 2: Subproblem minimization. Set a trust-region radius Δ_k that define the trust-region

$$B_k = \{x_k + s : s \in \mathbb{R}^n, \|s\| \leq \Delta_k\}$$

and minimize m_k in B_k.

Step 3: Accepting or rejecting the step. Compute the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s)}{m_k(x_k) - m_k(x_k + s)}.$$

If ρ_k is sufficiently positive, the iteration is successful: the next iteration point, $x_{k+1} = x_k + s$ will be taken and the trust-region radius Δ_{k+1} could be enlarged. If ρ_k is not positive enough, then the iteration was not successful: the current iterate x_k will be kept and the trust-region radius is reduced.

2.1 Interpolation ideas

To define the model $m_k(x_k + s)$ we need to obtain the vector g_k and the symmetric matrix G_k. They are both determined by requiring that the model m_k interpolates the function f at a set $Y = \{y^1, y^2, \ldots, y^p\}$ of points containing the current iterate x_k

$$f(y^i) = m_k(y^i) \quad \text{for all } y^i \in Y. \tag{1}$$

The cardinality of Y must be $p = \frac{1}{2}(n+1)(n+2)$ to get a full quadratic model m_k. Since there are $\frac{1}{2}(n+1)(n+2)$ coefficients to be determined in the model, the interpolation conditions represent a square system of linear equations in the coefficients d_k, g_k, G_k. If the interpolation points $\{y^1, y^2, \ldots, y^p\}$ are adequately chosen, the linear system is nonsingular and the model could be uniquely determined [14]. In practice, however, conditions about the geometry of the interpolation set (poisedness) are required in order to obtain a good model.

3 The NEWUOA and BOBYQA algorithms

NEWUOA is an algorithm proposed by Powell in [38] based on previous articles [34, 35, 36, 37]. This method has a sophisticated strategy in order to manage the trust-region radius and the radius of the interpolation set. The smaller of the two radii is used to force the interpolation points to be sufficiently far apart to avoid the influence of noise in the function values. Hence, the trust-region updating step is more complicated than the classical steps in the trust-region framework [14].

The main characteristic features of NEWUOA are the following:

(i) It uses quadratic approximations to the objective function aiming at a fast rate of convergence in iterative algorithms for unconstrained optimization. However, each quadratic model has $\frac{1}{2}(n+1)(n+2)$ independent coefficients to be determined, and this number could be prohibitively expensive in many applications with large n. So, NEWUOA tries to construct suitable quadratic models from fewer data. Each interpolation set has p points where $n+2 \leq p \leq \frac{1}{2}(n+1)(n+2)$. The default value in

NEWUOA is $p = 2n + 1$. Since p could be less than $\frac{1}{2}(n + 1)(n + 2)$, the interpolation set Y may not be complete. The remaining degrees of freedom are calculated by minimizing the Frobenius norm of the difference of two consecutive Hessian models [37]. This procedure defines the model uniquely, whenever the constraints (1) are consistent and $p \geq n + 2$, because the Frobenius norm is strictly convex. The updates along the iterations take advantage of the assumption that every update of the interpolation points is the replacement of just one point by a new one. Powell justified in [37], using the Lagrange polynomials of the interpolation points when one point x^+ replaces one of the points in Y, that it is possible to maintain the linear independence of the interpolation condition (1). It was shown that these conditions are inherited by the new interpolation points, when x^+ replaces y^t in Y, whenever they are chosen such that the Lagrange polynomial $L_t(x^+)$ is nonzero. Furthermore, the preservation of linear independence in the linear system (1) by the sequence of iterations, in the presence of computer rounding errors, may be more stable if $|L_t(x^+)|$ is relatively large.

(ii) It solves the trust-region quadratic minimization subproblem using a truncated conjugate gradient method. The Euclidean norm is adopted to define the trust-region B_k.

(iii) Updates of the interpolation set points are performed via the following steps:

(a) If the trust-region minimization of the k-th iteration produces a step s which is not too short compared to the maximum distance between the sample points and the current iterate, then the function f is evaluated at $x_k + s$ and this new point becomes the next iterate, x_{k+1}, whenever the reduction in f is sufficient. Furthermore, if the new point $x_k + s$ is accepted as the new iterate, it is included into the interpolation set Y, by removing the point y^t so that the distance $\|x_k - y^t\|$ and the value $|L_t(x_k + s)|$ are as large as possible. The trade-off between these two objectives is reached by maximizing the weighted absolute value $\omega_t|L_t(x_k + s)|$, where ω_t reflects the distance $\|x_k - y^t\|$.

(b) If the step s is rejected, the new point $x_k + s$ could be accepted into Y, by removing the point y^t such that the value $\omega_t |L_t(x_k + s)|$ is maximized, as long as either $|L_t(x_k + s)| > 1$ or $\|x_k - y^t\| > r\Delta_k$ is satisfied for a given $r \geq 1$.

(c) If the improvement in the objective function is not sufficient, and it is considered that the model needs to be improved, then the algorithm chooses a point in Y which is the farthest from x_k and attempts to replace it with a point which maximizes the absolute value of the corresponding Lagrange polynomial in the trust-region.

The general scheme of NEWUOA is the following:

Step 0: Initialization. Given x_0, NPT the number of interpolation points, Y the set of interpolation points, $\rho_0 > 0$, $0 < \rho_{end} < 0.1\rho_0$, $\Delta = \rho_0$, $\rho = \rho_0$, $t = 0$, $k \leftarrow 0$.
Build an initial quadratic model $m_0(x_0 + s)$ of the function $f(x)$.

Step 1: Solve the quadratic trust-region problem. Compute $\bar{s} = \text{argmin } m_k(x_k + s)$ subject to $\|s\| \leq \Delta$.

Step 2: Acceptance test. If $\|\bar{s}\| \geq 0.5\rho$, compute $ratio = (F(x_k) - F(x_k + \bar{s}))/(m_k(x_k) - m_k(x_k + \bar{s}))$.

Step 2.1: Update the trust-region radius. Reduce Δ and keep ρ. Set y^t the farthest interpolation point from x_k. If $\|y^t - x_k\| \geq 2\Delta$ go to Step 3, otherwise go to Step 5.

Step 2.2: Update the trust-region radius and ρ. Go to Step 6.

Step 3: Alternative iteration. Re-calculate $\bar{s}$ in order to improve the geometry of the interpolation set.

Step 4: Update the interpolation set and the quadratic model.

Step 5: Update the approximation. If $ratio > 0$, set $x_{k+1} \rightarrow x_k$, $k \rightarrow k + 1$ and go to Step 1.

Step 6: Stopping criterion. If $\rho = \rho_{end}$ declare "end of the algorithm", otherwise set $x_{k+1} \rightarrow x_k$, $k \rightarrow k + 1$ and go to Step 1.

The algorithm terminates if one of following options holds: the quadratic model does not decrease, $\rho = \rho_{end}$ or the maximum number of iterations is reached.

Numerical results of NEWUOA [39] encouraged the author to introduce some modifications in order to solve bound constrained optimization problems. The resulting algorithm is called BOBYQA [40].

In BOBYQA, Powell proposed the routine TRSBOX for solving the trust-region problem, obtaining an approximate solution of $m_k(x_k + s)$ in the intersection of a spherical trust-region with the bound constraints, via a truncated conjugate gradient algorithm. In BOBYQA we have replaced the routine TRSBOX by our version described below. Besides that, in the BOBYQA algorithm, others changes to the "variables" are designed to improve the model without reducing the objective function $f(x)$, using the subroutine ALTMOV. In that routine, the quadratic model $m_k(x_k + s)$ is ignored in the construction of a new $x_k + s$ by an *alternative* iteration. This subroutine (ALTMOV) is used by BOBYQA when the inclusion of the new iterate $x_k + s$, computed by TRSBOX, could determine a near linear dependence in the interpolation conditions. Thus, BOBYQA replaces the s of TRSBOX by a new s, computed by mean ALTMOV solving a quadratic function that is different to $m_k(x_k + s)$.

4 The active-set strategy for solving the box constrained subproblem

The quadratic minimization trust-region subproblem is one of the most expensive parts of the algorithm NEWUOA and BOBYQA. Both algorithms use the Euclidean norm to define the trust-region. In our case, we adopted the ∞-norm since we have bounds on the variables.

Given x_k and $\Delta_k > 0$ the current approximation and the trust-region radius, respectively, we define

$$B_k = \left\{ s \in \mathbb{R}^n \,|\, \|s\|_\infty \leq \Delta_k \right\}.$$

The trust-region subproblem is given by

$$\begin{array}{ll} \min & m_k(x_k + s) = \frac{1}{2} s^T G_k s + g_k^T s + d_k \\ \text{s. t.} & s \in \Omega_k = \Omega \cap B_k \end{array} \tag{2}$$

where $g_k \in \mathbb{R}^n$ and G_k is a $n \times n$ symmetric matrix.

In our algorithm, called TRB-Powell (Trust-Region Box), we replaced the quadratic solver of NEWUOA and BOBYQA by an active-set method that uses the strategy described in [7]. For solving the trust-region problem (2), this method uses a truncated Newton approach with line searches whereas, for leaving the faces, uses spectral projected gradient iterations as defined in [9]. Many active constraints can be added or deleted at each iteration so that the method is useful for large-scale problems. Besides that, numerical results have proved that this strategy is successful and efficient for medium and large-scale problems [2, 3, 6, 7, 8].

Specifically, suppose that s_j is the current iterate which belongs to a particular face of Ω_k. In order to decide if it is convenient to quit this face or to continue exploring it, we compute the gradient $\nabla m_k(s_j)$, its projection onto Ω_k (denoted by $g_P(s_j)$) and its projection onto this face ($g_I(s_j)$). Given $\eta \in (0, 1)$, if

$$\|g_I(s_k)\| \geq \eta \|g_P(s_k)\| \tag{3}$$

does not hold, the face is abandoned and s_{j+1} is computed performing one iteration using the spectral projected gradient algorithm, as it was described in [7]. On the other hand, while the condition (3) is satisfied, an approximate minimum of $m_k(x_k + s)$ belonging to the closure of Ω_k is computed using the conjugated gradient method. This procedure terminates when $\|g_P(s^*)\|$ is lower than a tolerance for some s^*.

The theoretical results in [7] allow us to assure that the application of this method to the quadratic model subject to Ω_k is well-defined and the convergence criterion is reached. In fact, Birgin and Martínez have proved that a Karush-Kuhn-Tucker (KKT) is computed up to an arbitrary precision. Also, assuming that all the stationary points are nondegenerate, the algorithm identifies the face to which the limit belongs in a finite number of iterations [7].

5 Numerical experiments

As mentioned in the introduction, we compared TRB-Powell with NEWUOA and BOBYQA in terms of number of function evaluations, as it is usual in derivative-free optimization articles. We have chosen two groups of test problems, small-size and medium-size problems. TRB-Powell was developed in

Fortran 77, as well as NEWUOA and BOBYQA. We have used Intel Fortran Compiler 9.1.036. Codes were compiled and executed in a PC running Linux OS, AMD 64 4200 Dual Core.

5.1 Test problems

Concerning the unconstrained case we considered a set of small-size problems proposed by Moré and Wild in [29]. Most of them consist of nonlinear least squares minimization problems from CUTEr collection [20]. The number of variables of these problems varies from 2 to 12. Also, aiming to test problems with larger dimensions, we considered several problems employed by Powell [38] where n varies from 20 to 100. These functions are Arwhead, Chrosen, Chebyqad, Extended Rosenbrock, Penalty1, Penalty2, Penalty3 and Vardim. All these functions are smooth and the respective results are showed in Table 3.

On the other hand, for bound constrained optimization we have considered some small-size problems from [22, 28, 29], whose names and dimensions can be seen in Table 4.

In order to test medium-size problems we have considered the Arwhead, Chebyqad, Penalty1 and Chrosen functions subject to bound constraints. The bound constraints of the first three problems require that all the components of x to be in the interval $[-10, 10]$ and in the case of Chrosen function in the interval $[-3, 0]$. Furthermore, we have considered one particular test problem studied in [40]. That problem, denominated "points in square (Invdist2)", has many different local minima. The objective function being the sum of the reciprocals of all pairwise distances between the points P_i, $i = 1, \ldots, M$ in two dimensions, where $M = n/2$ and where the components of P_i are $x(2i-1)$ and $x(2i)$. Thus, each vector $x \in \mathbb{R}^n$ defines the M points P_i. The initial point considered x_0 gives equally spaced points on a circle. The bound constraints of this problem require that all the components of x to be in the interval $[-1, 1]$.

The details of the results in [40] showed that they are highly sensitive to computer rounding errors. Other set of medium-size problems was taken from [28]. They are Ackley, Griewangk and Rastrigin functions, which have several local minima in each one particular search domain: $15 \leq x(i) \leq 30$, $-600 \leq x(i) \leq 600$, and $-5.12 \leq x(i) \leq 5.12$, $i = 1, \ldots, n$ respectively.

These functions have the global minimum at $x^* = 0$ and the corresponding value $f(x^*) = 0$.

Moreover, we have considered some problems of CUTEr that were used in the recent article by Gratton et al. [21], where the authors compared their new solver BC-DFO with BOBYQA. The detailed list of these problems and their characteristics is provided in Table 1. It shows the name and dimension of the problems, the number of variables which are bounded from below and above and the minimum value f^* which was reported in [21]. If one variable has not an upper or lower bound we used a very large bound for numerical considerations (10^{10} or -10^{10} respectively). The results of these experiments can be seen in Table 6.

5.2 Implementation details

We used the default parameters for codes NEWUOA and BOBYQA. We run both codes with a number $m = 2n + 1$ interpolation points using the Frobenius norm approach, as Powell suggest [38].

Initial points and initial trust-region radius were the same as in the cited references [21, 22, 28, 29, 38].

The stopping criterion that we have used is the same that Powell used, that is, the iterative process stops when the trust-region radius is lower than a tolerance $\rho_{end} = 10^{-6}$.

The maximum number of function evaluations allowed for the unconstrained case was:

$maxfun = 9000$, for small-size problems, and
$maxfun = 80000$, for medium-size problems.

The maximum number of function evaluations allowed for the bound constrained case was:

$maxfun = 9000$, for small-size problems, and
$maxfun = 20000$, for medium-size problems.

In the following tables the symbol $(**)$ indicates that the respective solver failed to find a solution or the maximum number of function evaluations allowed was reached.

Problem	n	lbound	ubound	(l+u)bound	f^*
Biggsb1	25			24	1.50000000000000D-02
Bqp1var	1			1	0.00000000000000D+00
Camel6	2			2	–1.03162845348988D+00
Chebyqad	4			4	2.56057805386809D-22
Chenhark	10	10			–2.00000000000000D+00
Cvxbqp1	10			10	2.47500000000000D+00
Explin2	12			13	–7.09247239439664D+03
Hatflda	4	4			1.61711062151584D-25
Hatfldc	25			24	3.43494690036517D-27
HS1	1	1			7.13660798093435D-24
HS110	10			10	–4.57784755318868D+01
HS2	2	1			4.94122931798918D+00
HS25	3			3	1.81845940377455D-16
HS3	2	1			1.97215226305253D-36
HS38	4			4	2.02675622883580D-28
HS3mod	2	1			0.00000000000000D+00
HS4	2	2			2.66666666400000D+00
HS45	5			5	1.00000000040000D+00
HS5	2			2	–1.91322295498104D+00
Logros	2	2			0.00000000000000D+00
Mccormck	10			10	–9.59800619474625D+00
Mdhole	2	1			7.52316384526264D-35
Ncvxbqp1	10			10	–2.20500000000000D+04
Ncvxbqp2	10			10	–1.43818650000000D+04
Oslbqp	8	5			6.25000000000000D+00
Probpenl	10			10	–2.11912948080046D+05
Pspdoc	4		1		2.41421356237309D+00
Qudlin	12			12	–7.20000000000000D+03
Simbqp	2			1	0.00000000000000D+00
Sineali	4			4	–2.83870492243045D+02

Table 1 – Bound constrained CUTEr test problems.

Algorithmic parameters in TRB-Powell (Step 1) used:
$\varepsilon_P = 10^{-6}$.
$\eta = 0.1$. We have analyzed other values, but the best results were obtained with this value of η.
$\gamma = 10^{-4}$, $\sigma_{\min} = 10^{-10}$ and $\sigma_{\max} = 10^{10}$.
In the conjugate gradient method we have used $\bar{\varepsilon} = 10^{-8}$.

5.3 Numerical results: unconstrained problems

Tables 2 and 3 report the name of the small and medium-size unconstrained optimization problems respectively, the number of function evaluations (Feval) used to reach the stopping criterion and the functional values obtained for TRB-Powell and NEWUOA codes ($f(x_{end})$).

The results in Table 2 enable us to make the following observations.

The number of function evaluations required by NEWUOA is the smallest in 16 of the 36 problems while for TRB-Powell this number is the smallest in 17 problems. In the rest of the problems, both algorithms performed the same number of function evaluations.

The average of the evaluations required in this whole set of problems was 1119 for TRB-Powell and 1272 for NEWUOA. In Bard problem NEWUOA got a local minimum while TRB-Powell obtained the global minimum. The obtained functional values are similar for both methods. NEWUOA obtained lower functional values in 18 problems whereas TRB-Powell did it in 14 of them.

These results are summarized in Figure 1 using the performance profiles described by Dolan and Moré in [16]. Given a set of problems P and a set S of optimization solvers, they compare the performance on problem $p \in P$ by a particular algorithm $s \in S$ with the best performance by any solver on this problem. Denoting by $t_{p,s}$ the number of function evaluations required when solving problem $p \in P$ by the method $s \in S$, they define the performance ratio:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

Problem		Feval		$f(x_{end})$	
name	**n**	**NEWUOA**	**TRB-Powell**	**NEWUOA**	**TRB-Powell**
Linear Full Rank	9	42	42	3.600000D+01	3.599990D+01
Linear Rank 1	7	159	308	8.380200D+00	8.380202D+00
Linear Rank 1*	7	151	225	9.880500D+00	9.880507D+00
Rosenbrock	2	195	134	1.382200D-17	1.089153D-16
Helical valley	3	124	146	2.066500D-13	4.180530D-14
Powell singular	4	476	477	1.099100D-10	1.350267D-10
Freudenstein & Roth	2	76	74	4.898420D+01	4.898420D+01
Bard	3	114	145	1.774900D+00	8.215013D-03
Kowalik & Osborne	4	278	274	3.075000D-04	3.075056D-04
Watson	6	937	1023	2.287600D-03	2.287600D-03
	9	9000	9000	(**) 2.160293D-06	(**) 1.782082D-06
	12	9000	9000	(**) 5.562963D-07	(**) 4.902148D-08
Box 3-dimensional	3	212	282	3.336000D-17	2.476166D-12
Jennrich & Sampson	2	55	65	1.243621D+02	1.243621D+02
Brown & Dennis	4	191	180	8.582220D+04	8.582220D+04
Chebyqad	6	177	212	4.763800D-13	1.298316D-14
	7	229	267	3.148000D-13	2.429278D-14
	8	390	247	3.516800D-03	3.516803D-03
	9	529	533	2.201700D-13	3.092237D-13
	10	666	535	4.772700D-03	4.772713D-03
	11	538	657	2.799700D-03	2.799751D-03
Brown almost-linear	10	1255	1153	1.553300D-12	1.588711D-12
Osborne 1	5	1012	1000	6.745600D-05	6.869259D-05
Osborne 2	11	1709	1024	4.013700D-02	4.013741D-02
Bdqrtic	8	432	507	1.023890D+01	1.023897D+01
	10	670	522	1.828100D+01	1.828118D+01
	11	758	614	2.226000D+01	2.226062D+01
	12	781	636	2.627200D+01	2.627277D+01
Cube	5	2842	2050	7.617000D-07	1.277429D-07
	6	4625	3080	4.772300D-06	3.328595D-06
	8	6825	4580	5.613800D-06	4.744310D-06
Mancino	5	39	55	3.712400D-11	9.547387D-14
	8	52	68	1.013600D-08	1.067564D-10
	10	68	67	2.292100D-09	1.024540D-08
	12	83	89	1.216000D-08	9.571743D-10
Heart 8	8	1118	1001	1.516400D-11	5.108511D-13

*with zero columns and rows

Table 2 – Small-size unconstrained problems: NEWUOA *vs.* TRB-Powell.

Problem		Feval		$f(x_{end})$	
name	**n**	**NEWUOA**	**TRB-Powell**	**NEWUOA**	**TRB-Powell**
Arwhead	20	409	495	6.827872D-12	3.321787D-13
	40	1441	1079	6.320278D-12	1.322941D-12
	80	3226	2791	8.715428D-11	3.022693D-11
	100	3859	3705	4.438583D-11	4.041300D-11
Penalty 1	20	7348	6768	1.577771D-04	1.577771D-04
	40	14620	18217	3.392511D-04	3.392511D-04
	80	31180	20037	7.130502D-04	7.131754D-04
	100	39364	22341	9.024910D-04	9.027157D-04
Penalty 2	20	19060	11180	6.389751D-03	6.389759D-03
	40	15301	11889	5.569125D-01	5.569250D-01
	80	19357	16863	1.776315D+03	1.776315D+03
	100	15388	12523	9.709608D+04	9.709608D+04
Penalty 3	20	4660	1645	3.636060D+02	1.008361D-02
	40	80000	5423	(**) 1.045973D-03	1.000000D-03
	80	80000	33231	(**) 6.285525D+03	1.000000D-03
	100	80000	78017	(**) 9.882811D+03	4.658901D-02
Chrosen	20	911	1005	2.031934D-11	2.128075D-13
	40	2048	2329	4.950602D-12	5.195116D-12
	80	4764	4529	1.803505D-10	5.955498D-11
	100	4987	5814	4.664747D-10	1.350874D-10
Vardim	20	4791	6449	5.365843D-11	6.547327D-12
	40	18725	28972	7.200291D-11	6.995020D-12
	80	62562	46687	4.743123D-10	1.196846D-09
	100	80000	77995	(**) 2.408154D-08	9.065575D-08
Rosenbrock Ext.	20	8585	9301	1.491622D-10	5.035342D-12
	40	36435	30310	7.109087D-09	3.092178D-12
	80	80000	78360	(**) 1.131952D-07	2.712245D-08
	100	80000	52891	(**) 2.812546D-07	2.572285D-09
Chebyqad	20	1817	2177	4.572955D-03	4.572955D-03
	40	26743	19970	5.960843D-03	5.961027D-03
	80	73006	30800	4.931312D-03	4.931938D-03
	100	46964	35200	8.715769D-03	4.517172D-03

Table 3 – Medium-size unconstrained problems: NEWUOA *vs.* TRB-Powell.

They assume that $r_{p,s} \in [1, r_M]$ and that $r_{p,s} = r_M$ only when problem p is not solved by solver s. They also define the fraction

$$\varrho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\},$$

where n_p is the number of solved problems. Thus, we draw $\varrho_s(\tau)$, with $\tau \in [1, r_M)$.

In a performance profile plot, the top curve represents the most efficient method within a factor τ of the best measure. The percentage of the test problems for which a given method is best in regard to the performance criterion being studied is given on the left axis of the plot. It is necessary to point out that when both methods coincide with the best result, both receive the corresponding mark. This means that the sum of the successful percentages may exceed 100%.

Figure 1 shows the performance profile for both solvers in the interval [1, 1.97]. It can be seen that NEWUOA performs less function evaluations in 52% of the problems while TRB-Powell does it in 55% of problems and the last one has the best performance for $\tau \in [1.18,\ 1.58]$.

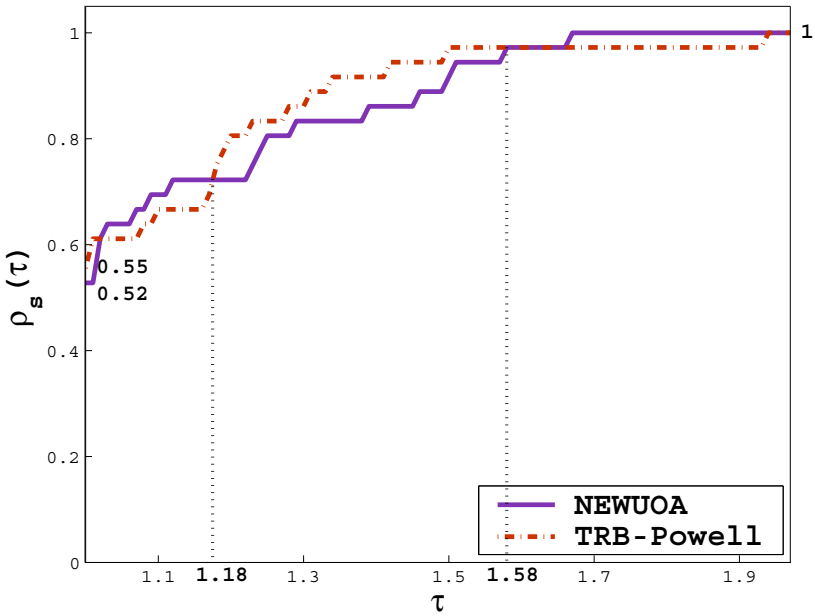

Figure 1 – Function evaluations in small-size unconstrained problems.

For medium-scale problems we tested eight problems from [38], with dimension $n = 20, 40, 80, 100$. Table 3 reports the numerical results, from which we can observe:

TRB-Powell required less function evaluations than NEWUOA in 23 of the 32 problems.

The average of the evaluations made by TRB-Powell was 21219 and for

NEWUOA was 29611. For eight problems TRB-Powell performed better than NEWUOA (NEWUOA needs more than 10000 extra function evaluations).

In Arwhead, Penalty 3, Rosenbrock Extended, Chrosen (n = 20) and Chrosen (n=80) problems, it can be seen that TRB-Powell was more efficient respect to the final functional value obtained ($f(x_{end})$). In particular, the Penalty 3 was solved significantly better by TRB-Powell than NEWUOA.

TRB-Powell obtained lower functional values than NEWUOA in 18 of 32 problems, while NEWUOA did it in 9. In the rest of the problems both methods achieve the same functional values.

This seems to indicate that for medium-size problems TRB-Powell performs better than NEWUOA.

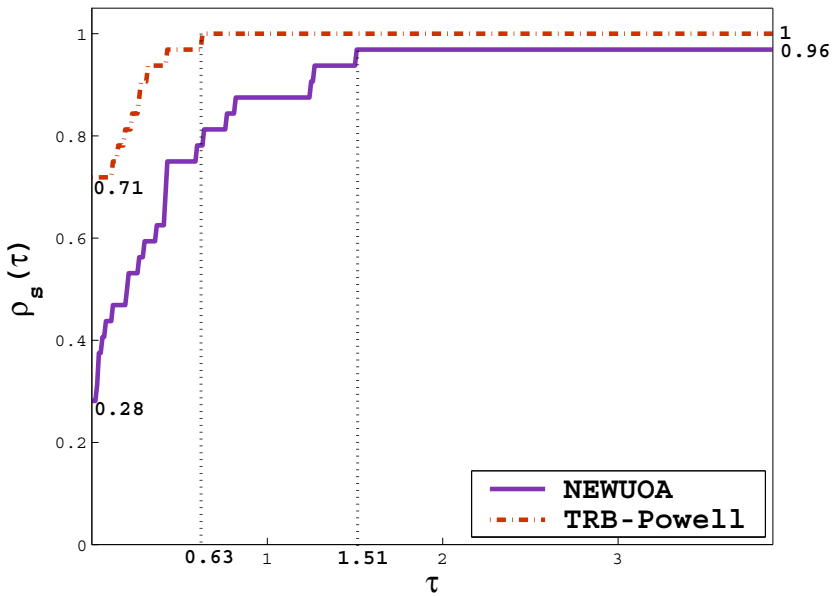

Figure 2 – Function evaluations in medium-size unconstrained problems.

These results are summarized in Figure 2 using the performance profiles described before. Since $r_{p,s}$ is large for several medium-size problems, we used a logarithmic scale in base 2 in the x-axis, as recommended in [16]. Thus, we draw

$$\varrho_s(\tau) = \frac{1}{n_p} size\{p \in P : \log_2(r_{p,s}) \leq \tau\}, \quad \text{with} \quad \alpha \in [0, \log_2(r_M)),$$

Problem		Feval		$f(x_{end})$	
name	**n**	**BOBYQA**	**TRB-Powell**	**BOBYQA**	**TRB-Powell**
Linear Full Rank	9	31	31	1.699999D+02	1.699999D+02
Linear Rank 1	7	84	82	8.380281D+00	8.380281D+00
Linear Rank 1*	7	90	82	9.880597D+00	9.880597D+00
Rosenbrock	2	171	170	4.720012D-12	4.213517D-09
Powell singular	4	131	83	4.840000D-04	4.840000D-04
Kowalik & Osborne	4	162	90	4.470176D-04	4.470176D-04
Brown Dennis	4	170	168	8.582220D+04	8.582220D+04
HS45	5	44	43	1.000000D+00	1.000000D+00
HS46	5	49	38	4.930380D-32	4.930380D-32
HS56	7	119	98	1.052166D-12	2.148472D-12
HS77	5	140	108	2.130995D-11	2.424337D-11
HS79	5	128	133	8.368490D-13	1.026634D-12
HS107	9	195	158	5.909464D-13	3.809796D-13
HS110	10	206	172	-4.577847D+01	-4.577847D+01
Rastrigin	10	69	68	2.487372D+02	2.487372D+02
Schwefel	10	90	73	-4.189829D+03	-4.189829D+03

*with zero columns and rows

Table 4 – Small-size bounded constrained problems: BOBYQA *vs.* TRB-Powell.

where $r_M > 0$ is such that $r_{p,s} \leq r_M$, for all p and s.

Figure 2 shows that TRB-Powell solved this set of problems successfully. TRB-Powell had a better performance in 71% of the problems.

5.4 Numerical results: bound constrained problems

Tables 4 and 5 show the performance of BOBYQA and TRB-Powell for small and medium-size bound constrained problems, respectively. We observe in Table 4 that the number of function evaluations required by TRB-Powell is less than BOBYQA in 14 of the 16 test problems. Moreover, the obtained functional values were similar for both methods. This performance of TRB-Powell has been depicted in Figure 3, where it has required less function evaluations in 93% of the problems.

On the other hand, for medium-size bound constrained problems, we observe in Table 5 that TRB-Powell required less function evaluations than BOBYQA

Problem		Feval		$f(x_{end})$	
name	**n**	**BOBYQA**	**TRB-Powell**	**BOBYQA**	**TRB-Powell**
Invdist2	20	182	236	3.220305D+01	3.220305D+01
	40	943	826	1.677709D+02	1.677709D+02
	60	6990	4832	4.081079D+02	4.087195D+02
	80	5158	6125	7.699010D+02	7.689744D+02
	100	20000	20000	(**) 1.251662D+03	(**) 1.248745D+03
Arwhead	20	636	558	1.826050D-11	4.961897D-11
	40	1458	1468	1.208688D-09	1.054593D-08
	80	3677	4332	8.622840D-10	1.097986D-09
	100	6605	2927	3.969482D-08	2.774640D-09
Chrosen	20	242	235	1.900000D+01	1.900000D+01
	40	464	612	3.900000D+01	3.900000D+01
	80	864	825	7.900000D+01	7.900000D+01
	100	1055	1037	9.900000D+01	9.900000D+01
Penalty 1	20	5677	5253	1.577706D-04	1.577901D-04
	40	15006	11645	3.392511D-04	3.392517D-04
	80	19410	15680	7.130502D-04	7.130518D-04
	100	20000	17458	(**) 9.024911D-04	9.024942D-04
Chebyqad	20	2245	2020	4.572955D-03	4.572956D-03
	40	20000	19617	(**) 5.960849D-03	5.960854D-03
	80	20000	20000	(**) 4.932619D-03	(**) 4.933963D-03
	100	20000	20000	(**) 4.523622D-03	(**) 4.566568D-03
Rastrigin	20	120	114	4.974745D+02	4.974745D+02
	40	1301	297	9.949489D+02	9.949489D+02
	80	600	613	1.989898D+03	1.989898D+03
	100	820	795	2.487372D+03	2.487372D+03
Griewangk	20	882	523	6.312884D-11	3.234968D-12
	40	2091	1309	7.525736D-11	1.494938D-11
	80	12436	3031	2.297593D-10	1.809503D-10
	100	5213	3927	2.405037D-10	6.107597D-10
Ackley	20	570	526	9.132573D-06	9.355251D-07
	40	1129	1231	9.987129D-07	1.001365D-06
	80	2112	1923	9.156053D-06	9.086830D-06
	100	2968	2406	8.297680D-07	9.516777D-06

Table 5 – Medium-size bounded constrained problems: BOBYQA *vs.* TRB-Powell.

in 23 of the 33 test problems. Besides that, our method compares well with BOBYQA in the sense of TRB-Powell needed less function evaluations. The final functional values were similar for BOBYQA and TRB-Powell. Figure 4 shows that TRB-Powell solved the problems using less function evaluations in 78% of them.

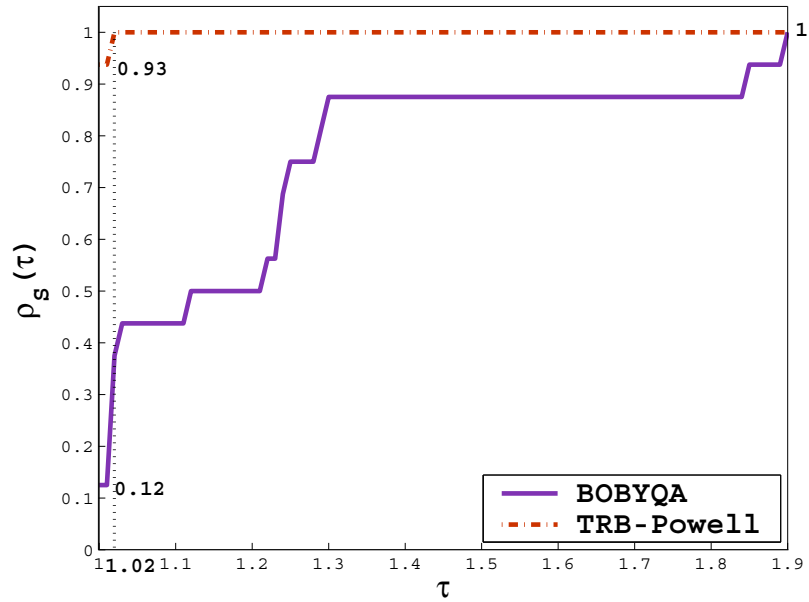

Figure 3 – Function evaluations in small-size bound constrained problems.

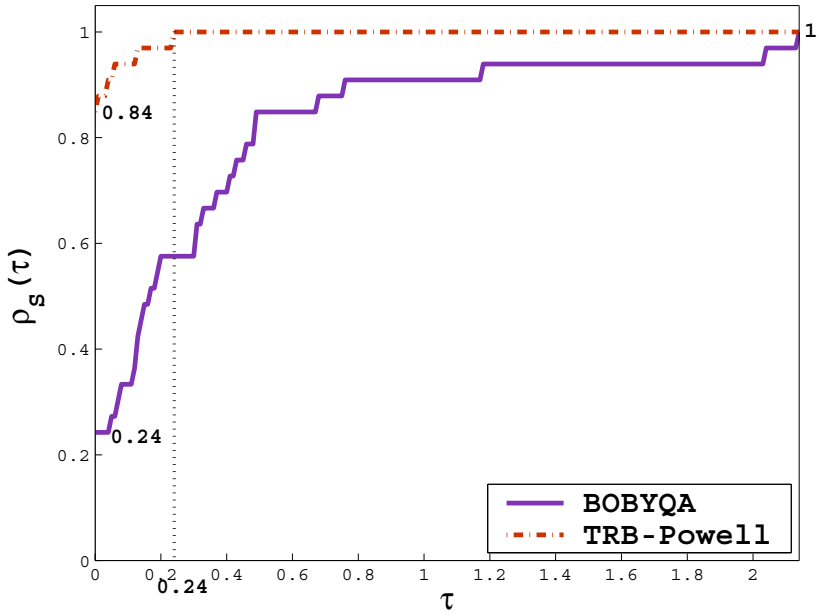

Figure 4 – Function evaluations in medium-size bound constrained problems (log scale).

Problem		Feval							
name	n	2-dig		4-dig		6-dig		8-dig	
		TRB	BQA	TRB	BQA	TRB	BQA	TRB	BQA
Biggsb1	25	107	144	315	341	389	489	464	548
Bqp1var	1	4	7	5	7	5	7	5	7
Camel6	2	15	17	25	29	29	37	31	41
Chebyqad	4	10	15	18	50	28	60	48	64
Chenhark	10	108	90	124	121	143	151	161	172
Cvxbqp1	10	30	26	30	39	30	39	30	39
Explin2	12	126	224	210	233	219	236	227	258
Hatflda	4	36	46	152	104	167	141	180	182
Hatfldc	25	56	131	87	247	196	360	276	441
HS1	2	102	135	128	158	145	167	158	172
HS110	10	23	144	44	260	82	436	188	521
HS2	2	40	33	44	35	44	39	44	39
HS25	3	97	107	710	734	968	978	f	995
HS3	2	9	6	9	9	9	10	9	10
HS38	4	375	408	431	440	468	474	501	503
HS3mod	2	16	21	21	24	21	24	21	24
HS4	2	6	7	7	7	7	7	7	7
HS45	5	16	16	16	16	16	16	16	16
HS5	2	9	13	12	15	17	18	22	21
Logros	2	82	443	399	609	445	652	547	661
Mccormck	10	40	29	82	54	83	75	99	87
Mdhole	2	218	220	220	220	227	225	228	225
Ncvxbqp1	10	28	31	28	31	28	31	28	31
Ncvxbqp2	10	28	31	28	31	28	31	28	31
Oslbqp	8	20	22	20	22	20	22	20	22
Probpenl	10	f	f	f	f	f	f	f	f
Pspdoc	4	40	41	50	55	57	57	65	65
Qudlin	12	32	34	32	34	32	34	32	34
Simbqp	2	12	12	12	12	12	12	12	12
Sineali	4	12	36	17	260	17	387	17	432

Table 6 – Results on bound constrained CUTEr test set with 2, 4, 6 and 8 significant figures attained in f^*.

Table 6 shows the results obtained by BOBYQA (BQA) and TRB-Powell (TRB) on the experiments with the problems described in Table 1. It shows the name of the problem, the number of function evaluations needed by TRB-Powell and BOBYQA to attain two, four, six and eight significant figures of the objective function value f^*, reported by the authors of [21]. We indicate with "f" when a problem do not obtain the precision required.

The results reported in Table 6 show that both solvers fail to solve one test problem in all four cases. Moreover, TRB-Powell did not obtain the precision of 8 digits in the HS25 problem. For low accuracy BOBYQA solved 17% of the test cases faster than TRB-Powell and TRB-Powell solved 73% of the problems faster. For 8 correct significant figures, BOBYQA solved 17% of the test cases faster, and TRB-Powell 67% of the problems faster. For 4 and 6 significant digits BOBYQA solved 13% of the test cases faster than TRB-Powell and TRB-Powell solved 67% and 70% of the problems faster, respectively.

6 Conclusions

We have presented a modified version of the algorithms NEWUOA and BOBYQA for solve unconstrained and bound constrained derivative-free optimization problems. Our method use an active-set strategy for solving the trust-region subproblems. Since we consider the infinity norm, a box constrained quadratic optimization problem has to be solved at each iteration.

We have compared our new version TRB-Powell with the original NEWUOA and BOBYQA. The numerical results reported in this paper are encouraging and suggest that our algorithm takes advantage of the active-set strategy to explore the trust-region box. The number of function evaluations was reduced in most of the cases. These promising numerical results and the new articles by M.J.D. Powell [39, 40] encourage us for further development in constrained optimization without derivatives.

Acknowledgements. We are indebted to two anonymous referees whose comments helped a lot to improve this paper.

REFERENCES

[1] P. Alberto, F. Nogueira, H. Rocha and L. Vicente, *Pattern search methods for user-provided points: Application to molecular geometry problems*. SIAM Journal on Optimization, **14**(4) (2004), 1216–1236.

[2] R. Andreani, E. Birgin, J.M. Martínez and M.L. Schuverdt, *On augmented lagrangian methods with general lower-level constraints*. SIAM Journal on Optimization, **18** (2007), 1286–1309.

[3] R. Andreani, E. Birgin and J.M. Martínez and M.L. Schuverdt, *Augmented lagrangian methods under the constant positive linear dependence constraint qualification*. Mathematical Programming, **111** (2008), 5–32.

[4] C. Audet and J. Dennis Jr., *A pattern search filter method for nonlinear programming without derivatives*. SIAM Journal on Optimization, **14**(4) (2004), 980–1010.

[5] F. Berghen and H. Bersini, *CONDOR: a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm*. Journal of Computational and Applied Mathematics, **181**(1) (2005), 157–175.

[6] E. Birgin, R. Castillo and J.M. Martínez, *Numerical comparison of augmented lagrangian algorithms for nonconvex problems.* Computational Optimization and Applications, **31** (2005), 31–56.

[7] E. Birgin and J.M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients.* Computational Optimization and Applications, **23**(1) (2002), 101–125.

[8] E. Birgin and J.M. Martínez, *Improving ultimate convergence of an augmented lagrangian method.* Optimization Methods and Software, **23** (2008), 177–195.

[9] E. Birgin, J.M. Martínez and M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets.* SIAM Journal on Optimization, **10** (2000).

[10] A. Booker, J. Dennis Jr., P. Frank, D. Serafini and V. Torczon, *Optimization using surrogate objectives on a helicopter test example*. Computational Methods for Optimal Design and Control, J. Borggaard, J. Burns, E. Cliff and S. Schreck eds. Birkhäuser, (1998), 49–58.

[11] A. Conn, K. Scheinberg and P. Toint, *On the convergence of derivative-free methods for unconstrained optimization.* Approximation Theory and Optimization: Tributes to M. Powell, M. Buhmann and A. Iserles Eds., Cambridge University Press, Cambridge, UK (1997), 83–108.

[12] A. Conn, K. Scheinberg and P. Toint, *Recent progress in unconstrained nonlinear optimization without derivatives.* Mathematical Programming, **79** (1997), 397–414.

[13] A. Conn, K. Scheinberg and P. Toint, *A derivative free optimization algorithm in practice.* In Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO (1998).

[14] A. Conn, K. Scheinberg and L. Vicente, *Introduction to derivative-free optimization.* SIAM (2009).

[15] J.E. Dennis and V. Torczon, *Direct search methods on parallel machines.* SIAM Journal on Optimization, **1** (1991), 448–474.

[16] E. Dolan and J. Moré, *Benchmarking optimization software with performance profiles.* Mathematical Programming, **91** (2002), 201–213.

[17] R. Duvigneau and M. Visonneau, *Hydrodynamic design using a derivative-free method.* Structural and Multidisciplinary Optimization, **28**(2) (2004), 195–205.

[18] G. Fasano, J. Morales and J. Nocedal, *On the geometry phase in model-based algorithms for derivative-free optimization*. Optimization Methods and Software, **24**(1) (2009), 145–154.

[19] K. Fowler, J. Reese, C. Kees, J. Dennis Jr., C. Kelley, C. Miller, C. Audet, A. Booker, G. Couture, R. Darwin, M. Farthing, D. Finkel, J. Gablonsky, G. Gray and T. Kolda, *A comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems.* Advances in Water Resources, **31**(5) (2008), 743–757.

[20] N.I. Gould, D. Orban and Ph.L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited.* ACM Transactions on Mathematical Software, **29**(4) (2003), 373–394.

[21] S. Gratton, Ph.L. Toint and A. Tröltzsch, *An active-set trust region method for derivative-free nonlinear bound-constrained optimization.* Technical Report. CERFACS. Parallel Algorithms Team., **10** (2010), 1–30.

[22] W. Hock and K. Schittkowski, *Nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems, Springer, **187** (1980).

[23] T. Kolda, R. Lewis and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods.* SIAM Review, **45**(3) (2003), 385–482.

[24] T. Levina, Y. Levin, J. McGill and M. Nediak, *Dynamic pricing with online learning and strategic consumers: An application of the aggregation algorithm*. Operations Research (2009), 385–482.

[25] M. Marazzi and J. Nocedal, *Wedge trust region methods for derivative free optimization.* Mathematical Programming, **91**(2) (2002), 289–305.

[26] A. Marsden, M. Wang, J. Dennis and P. Moin, *Optimal aeroacustic shape design using the surrogate management framework.* Optimization and Engineering, **5**(2) (2004), 235–262.

[27] J. Meza and M.L. Martínez, *On the use of direct search methods for the molecular conformation problem.* J. Comput. Chem., **15**(6) (1994), 627–632.

[28] M. Molga and C. Smutnicki, *Test functions for optimization needs.* Available at http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf (2005).

[29] J. Moré and S. Wild, *Benchmarking derivative-free optimization algorithms.* SIAM Journal on Optimization, **20**(1) (2009), 172–191.

[30] P. Mugunthan, C. Shoemaker and R. Regis, *Comparison of function approximation, heuristic and derivative-based methods for automatic calibration fo computationally expensive groundwater bioremediation models.* Water Resour. Res., **41**(11) (2005), 1–17.

[31] J. Nelder and R. Mead, *A simplex method for function minimization.* Computer Journal, **7** (1965), 308–313.

[32] R. Oeuvray, *Trust-region methods based on radial basis functions with application to biomedical imaging.* Ph.D. thesis (2005).

[33] R. Oeuvray and M. Bierlaire, *A new derivative-free algorithm for the medical image registration problem.* International Journal of Modelling and Simulation, 27(2) (2007), 115–124.

[34] M.J.D. Powell, *On the Lagrange functions of quadratic models that are defined by interpolation.* Optimization Methods and Software, **16** (2001), 289–309.

[35] M.J.D. Powell, *UOBYQA: Unconstrained optimization by quadratic approximation.* Mathematical Programming, **92**(3) (2002), 555–582.

[36] M.J.D. Powell, *On trust region methods for unconstrained minimization without derivatives.* Mathematical Programming, **97**(3) (2003), 605–623.

[37] M.J.D. Powell, *Least Frobenius norm updating of quadratic models that satisfy interpolation conditions.* Mathematical Programming, **100**(1) (2004), 183–215.

[38] M.J.D. Powell, *The NEWUOA software for unconstrained optimization without derivatives.* Nonconvex Optimization and Its Applications, Springer US, **83** (2006).

[39] M.J.D. Powell, *Developments of NEWUOA for minimization without derivatives.* IMA Journal of Numerical Analysis, **28**(4) (2008), 649–664.

[40] M.J.D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives.* Technical report, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England (2009).

[41] V. Torczon, *On the convergence of pattern search algorithms.* SIAM Journal on Optimization, **7**(1) (1997), 1–25.

[42] D. Winfield, *Functions and functional optimization by interpolation in data tables.* Ph.D. thesis (1969).

[43] D. Winfield, *Functions minimization by interpolation in a data table.* IMA Journal of Applied Mathematics, **12**(3) (1973), 339–347.