UNIVERSIDAD NACIONAL DE CÓRDOBA

FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA

SERIE "A"

TRABAJOS DE MATEMÁTICA

Nº 96/09

Active-set strategy in NEWUOA for optimization without derivatives

María B. Arouxet - Nélida Echebest - Elvio A. Pilotta

Editores: Jorge R. Lauret – Jorge G. Adrover

CIUDAD UNIVERSITARIA – 5000 CÓRDOBA
REPÚBLICA ARGENTINA

Active-set strategy in NEWUOA for optimization without derivatives

María Belén Arouxét* Nélida Echebest * Elvio A. Pilotta †

August 24, 2009

Abstract

We present an algorithm for solving unconstrained optimization problems without derivatives based on NEWUOA, the Powell's algorithm, for derivative-free optimization. At each iteration, a quadratic interpolation model of the objective function around the current iterate is constructed and this model is minimized to obtain the new trial point. The whole process is embedded within a trust-region framework. In our algorithm, we use the infinity norm and we solve a box constrained quadratic problem by using an active set strategy exploring the faces of the box. An spectral gradient approach is used to abandon the face. Numerical experiments show that our algorithm requires less functional evaluations than NEWUOA, thus it confirms the promising behavior of the algorithm.

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

1 Introduction

We consider the unconstrained optimization problem where the derivatives of the objective function f are not available and the functional values $f(x)$ are typically very expensive or difficult to compute. That is, we consider the problem

$$\min f(x) \text{ subject to } x \in \mathbb{R}^n$$

where we assume that the function f is smooth enough and that $\nabla f(x)$ cannot be computed for any x.

*Departamento de Matemática Aplicada, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, Argentina ({belen|opti@mate.unlp.edu.ar}).

†Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, CIEM (CONICET), Medina Allende s/n, Ciudad Universitaria (5000) Córdoba, Argentina, (pilotta@famaf.unc.edu.ar)
Work sponsored by CONICET and SECYT-UNC.

This situation use to occur frequently in problems where the functional values $f(x)$ come from physical, chemical or geophysical measure or are the results from very complex computer simulations, which is common in problems from industry and engineering.

There are several, essentially, different methods for solving this kind of problems. See [8]. A first class of them are the direct search or pattern search methods which are based on the exploration of the variables space by using function evaluations in sample points given by a predefined geometric pattern. That is the case of methods where sampling is guided by desirable sets of directions [9, 28] and those based on simplices and operations over simplices like the Nelder-Mead algorithm [18]. They do not exploit the inherit smoothness of the objective function and require therefore a very large number of function evaluations. They can be useful for non-smooth problems. A comprehensive survey of these methods can be found in [15]. A second class includes those methods that use line-search strategies based on simplex derivatives, connecting with the implicit-filtering method [14]. The last class of the methods are based on modelling the objective function by multivariate interpolation in combination with the trust-region techniques. These methods were introduced by Winfield [29, 30]. The main idea of these methods is to build a polynomial model, interpolating the objective function at all points at which its value is known. The model is then minimized over the trust region and a new point, is computed. The objective function evaluated at this new point thus possibly enlarging the interpolation set. This newly computed point is checked as to whether the objective function is improved and the whole process repeated until convergence is achieved. So, the geometry of the interpolation set points and the model minimization are the keys for a good performance of the algorithms.

At the present time, there are several implementations of algorithms based on trust-region interpolation approach, although the most tested and well established are DFO developed by Conn, Scheinberg and Toint [5, 6, 7] and NEWUOA developed by Powell [20, 21, 22, 23, 24]. See also the Wedge method developed by Marazzi and Nocedal [16] and the CONDOR implementation of Berghen and Bersini [2] that is parallel version based on the Powell method.

For our proposal we used the model-based trust region method NEWUOA because this code performed very well in recent comparison benchmark articles by [19, 11, 17]. Moreover Moré and Wild [17] report that NEWUOA is the most effective derivative free optimization method for smooth function. These results and the recent developments by Powell [25] encourage us for further development of model based methods.

In this paper we decided to intervene in the trust-region subproblem solved by a truncated conjugate gradient method in NEWUOA. Instead of this solver, we used an active-set strategy [3] together with the spectral projected gradient method (SPG) developed by Martínez, Birgin and Raydán [4]. SPG combines a nonmonotone line search [13] with the spectral gradient method proposed by Barzilai and Borwein [1] and revisited by Raydán [26, 27].

The numerical results and the observations made in this paper are based on experiments involving all the smooth problems suggested in [17] and also we tested with a set of medium-scale problems (100 variables).

This article is organized as follows. The main interpolation-based methods ideas for deriva-

tive free optimization are given in Section 2. In Section 3 we give a short description of the NEWUOA solver for derivative free optimization. The active-set strategy and the spectral projected gradient method are described in Section 4. In Section 5 we show numerical experiments using a set of test problems. Also, we report numerical experiments by using some medium-scale problems and we make some comments related to numerical behavior. Conclusions are given in Section 6.

2 Main interpolation-based methods ideas for optimization without derivatives

The idea of locally approximate a function by using a cheaper model of the objective function is an usual globalization strategy in nonlinear optimization called trust-region. Basically, the main steps of the trust region method are the following:

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):

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

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

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.

3. Accept or reject the step. If the ratio of the achieved reduction in the objective function versus the predicted reduction in model

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

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

2.1 Interpolation ideas

To define the model in Step 1 we need to obtain the vector g_k and the symmetric matrix H_k. They are determined by requiring that the model m_k interpolates the function f at a set $Y_k = \{y^1, y^2, \ldots, y^q\}$ of points containing the current iterate x_k

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

The cardinality of Y_k must be

$$q = \frac{1}{2}(n+1)(n+2)$$

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 c_k, g_k, G_k. If the interpolation points $\{y^1, y^2, \ldots, y^q\}$ are adequately chosen the linear system is nonsingular and the model could be uniquely determined. See [8]. However, in practice the interpolation conditions are not sufficient to uniquely determine the solution and to guarantee the good quality of the model. Geometric conditions (poisedness) on the interpolation set are required to ensure the existence and uniqueness of the linear system solution.

3 The NEWUOA algorithm

NEWUOA is an algorithm proposed by Powell in (see [24]) based in previous articles [20, 21, 22, 23]. In fact, NEWUOA is the new version of UOBYQA [21]. The Powell's method has a complicated mechanism to manage the trust-region radius and the radius of the interpolation set. The smaller of the two radii is also used to force the interpolation points to be sufficiently far apart to avoid the influence of noise in the function values. The trust-region updating step is more complicated that the classical steps in trust-region framework [8]. These additional schemes are developed to improve the practical performance of the algorithms. They not affect the main global convergence results but make the analysis quite complex.

The main distinguishing features of the NEWUOA are the following:

- It uses quadratic approximations to the objective function which are highly useful for obtaining 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. Therefore NEWUOA tries to construct suitable quadratic models from fewer data. Each one interpolation sets 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 could not be complete. The remaining degrees of freedom are calculated by minimizing the Frobenius norm of the change in the model Hessian, with respect to the model used in the previous iteration.

- It uses Lagrangian polynomials to interpolate the function f by using the interpolation set Y.

- It solves the trust-region quadratic minimization subproblem by using truncated conjugate gradient methods. To define the trust-region B_k is used the Euclidean norm.

- Updates of the interpolation set points are performed via the following steps:

    - The set of minimum Frobenius norm Lagrange polynomials is maintained in every iteration.
    - If the trust-region minimization of the kth iteration produces a step s_k 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 $x_k + s_k$ and the new point becomes the next iterate, x_{k+1}, if the reduction in f is sufficient. If the new point $x_k + s_k$ is accepted as the new iterate, it is included into Y_k, by removing the point y^i such that the distance $\|x_k - y^i\|$ and the value $|l_i(x_k + s_k)|$ are as large as possible. The trade off between these two objectives is reached by maximizing the weighted absolute value $\omega_i |l_i(x_k + s_k)|$, where ω_i reflects the distance $\|x_k - y^i\|$.
    - When the step s_k is rejected, the new point $x_k + s_k$ can be accepted into Y_k, by removing the point y^i such that the value $\omega_i |l_i(x_k + s_k)|$ is maximized, where ω_i reflects the distance $\|x_k - y^i\|$, as long as either $|l_i(x_k + s_k)| > 1$ or $\|x_k - y^i\| > r\Delta_k$, for a given $r \geq 1$.
    - If the improvement in the objective function is not sufficient, and it is believed that the model needs to be improved, then the algorithm chooses a point in Y_k which is the furthest 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.

4 The active-set strategy with spectral projected gradient method

The quadratic minimization trust-region subproblem is one of the more expensive part of the algorithm. Therefore, we adopted the ∞-norm and decided to use another strategy in order to improve the numerical performance of NEWUOA. Our proposal is based on the box-constrained decomposition proposed by Friedlander and Martínez in [12].

The trust region subproblem is

$$\text{Minimize } m(x) \text{ subject to } x \in \Omega,$$

where $\Omega = \{x \in \mathbb{R}^n | l \leq x \leq u\}$.

As in [12], we divide the feasible set Ω into disjoint open faces, as follows. For all $I \subset \{1, 2, \ldots, n, n+1, n+2, \ldots, 2n\}$, we define

$$F_I = \{x \in \Omega \,|\, x_i = l_i \text{ if } i \in I, x_i = u_i \text{ if } n+i \in I,\ l_i < x_i < u_i \ \text{ otherwise } \}.$$

We also define V_I the smallest affine subspace that contains F_I and S_I, the parallel linear subspace to V_I. The (continuous) projected gradient at $x \in \Omega$ is defined as

$$g_P(x) = P_\Omega(x - g(x)) - x.$$

For all $x \in F_I$, we define the internal projected gradient

$$g_I(x) = P_{S_I}[g_P(x)]$$

that is, the projection of g_P into the parallel linear subspace S_I.

Now, for solving the trust-region quadratic subproblem we used the algorithms GENCAN and SPG. See [3].

Algorithm GENCAN.

Assume that $x_0 \in \Omega$ is an arbitrary initial point, $\eta \in (0, 1)$ and $0 < \sigma_{\min} < \sigma_{\max} < \infty$. Let F_I be the face that contains the current iterate x_k. Assume that $g_P(x_k) \neq 0$, otherwise the algorithms terminates. At the main iteration of the algorithm we test

$$\|g_I(x_k)\| \geq \eta \|g_P(x_k)\|. \tag{1}$$

If (1) is satisfied, we consider that is convenient that the new iterate belong to $\overline{F}_I$ (the closure of F_I) and we compute x_{k+1} by using the gradient conjugated method with the set of variables restricted to the free variables in F_I.

If (1) does not hold, we decide that some constraints should be abandoned and, so, the new iterate x_{k+1} is computed doing one iteration of the SPG Algorithm. In this case, before the computation of x_{k+1} we compute the spectral gradient coefficient σ_k in the following way:

If $k = 0$ or $(x_k - x_{k-1})^t(g(x_k) - g(x_{k-1})) \leq 0$ then

$$\sigma_k = \max\{1, \|x_k\|/\|g_P(x_k)\|\}.$$

Otherwise, define

$$\sigma'_k = \frac{(x_k - x_{k-1})^t(g(x_k) - g(x_{k-1}))}{\|x_k - x_{k-1}\|^2}$$

and

$$\sigma_k = \min\{\sigma_{\max}, \max\{\sigma_{\min}, 1/\sigma'_k\}\}.$$

Therefore, we use the following algorithm when it is necessary to leave the current face according to test (1).

Algorithm SPG.
Compute x_{k+1} as the next iterate of a monotone SPG iteration [4] with the spectral step σ_k. We define the search direction d_k as

$$d_k = P_\Omega(x_k - \sigma_k g(x_k)) - x_k$$

and we compute $x_{k+1} = x_k + \alpha_k d_k$ in such a way that

$$f(x_{k+1}) \leq f(x_k) + \gamma \alpha_k g(x_k)^t d_k,$$

trying first $\alpha_k = 1$ and, perhaps, reducing this coefficient by means of a safeguarded quadratic interpolation procedure.

5 Numerical experiments

In order to asses the performance of our algorithm (AS-NEW) we tested our implementation against NEWUOA using all the smooth problems from the test problems set in [17], where the number of variables varies from 2 to 12. Also, we consider a set of medium scale problems from [25]. NEWUOA was developed in Fortran 77, so the new algorithm too. We used Intel Fortran Compiler 9.1.036. Both codes were compiled and executed in a PC running Linux OS, AMD 64 4200 Dual Core.

Initial points and initial trust-region radius were the same as in the cited references [17, 25]. However, for some problems we also tested with the initial point $10x_s$, where x_s is the original initial point. The stopping criterion that we used is the same that Powell used in NEWUOA, that is, the iteration stopped when the trust-region radius is lower than a tolerance: $\rho_{end} = 10^{-6}$.

As it is usual in derivative free optimization articles we were interested in the comparison of the number of function evaluations for both codes. We also report the CPU time obtained for both algorithms. To compare them we considered the performance profiles methods described by Dolan and Moré in [10]. Formally, for each algorithm A_i there is distribution function ϱ_i. The real value $\varrho_i(\alpha)$ represents the probability that the ratio between the performance of A_i and the performance of the best algorithm is less than α. Then, $\varrho_i(1)$ represents the probability of algorithm A_i to perform as the best and $lim_{\alpha \to \infty^+} \varrho_i(\alpha)$ is a measure of the algorithm robustness. The more efficient will be that it reaches this limit rapidly. For derivative free optimization methods, as it is usual, the performance is measured by the number of functional evaluations to reach the convergence. In our numerical experiments, the maximum number of function evaluations allowed was:

- `MAX_FE = 8000,` for small size problems,
- `MAX_FE = 80000,` for medium size problems.

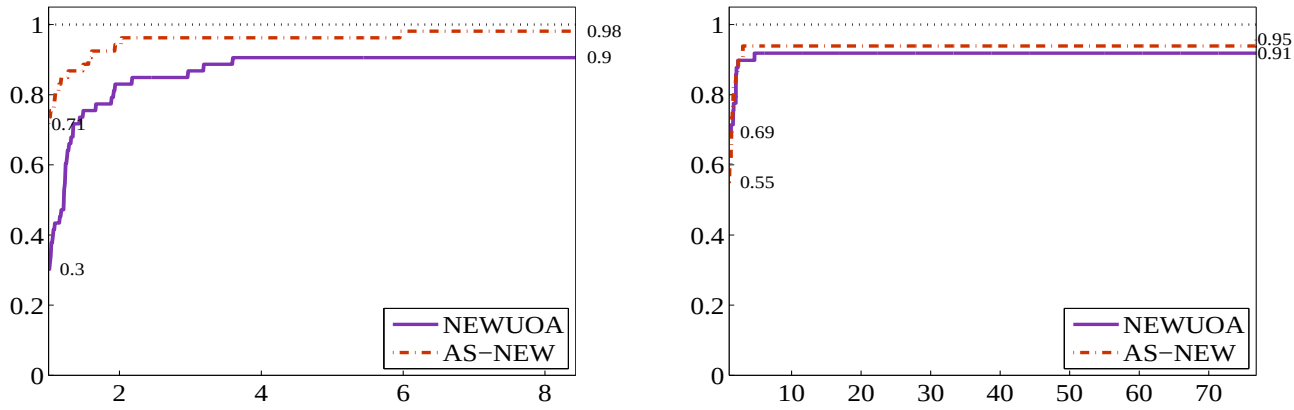

Figure 1: (a) Function evals.: NEWUOA vs. AS-NEW (b) CPU time: NEWUOA vs. AS-NEW

Table 1 reports the name of the small size problems, the initial points (I.P.), the number of function evaluations, the CPU time (in seconds) and the functional values obtained for both codes.

The results are summarized in Figures 1 using the performance profiles described in [10]. They compare the number of function evaluations and the CPU times (in seconds) needed to achieve the desired accuracy for both codes.

Remark that for $\alpha = 1$, Figure 1 (a) seems to indicate that AS-NEW is the best solver because the number of function evaluations was lower than NEWUOA on 71% of the problems. NEWUOA was better on 30% of the problems. For the remaining problems the number of function evaluations was identically for both codes. Also, it is worth to remark that AS-NEW solved successfully the 98% of the test problems while NEWUOA solved the 90% of them.

The comparison of CPU times, in Figure 1 (b), shows that both algorithms perform similarly in these tests, although NEWUOA run slightly faster than our implementation for some problems.

For medium scale problems we tested seven problems used by Powell in [25], where the dimension varies from 20 to 100. Table 2 reports the numerical results. We observe that for most of the problems the number of function evaluations using AS-NEW are remarkably lower than NEWUOA.

In general, the corresponding final functional values for both codes was similar in almost all problems. However, they differ for problems Arwhead and Vardim. We think that more investigation about the stopping criterion should be done.

We would like to mention that we run using different strategies (more or less conservative) for searching in face of the trust-region box, although the numerical results were similar.

As it can be seen in the plots of Figures 2, the performance of AS-NEW is surprisingly better than NEWUOA because AS-NEW solves successfully the 92% of the problems and perform less

Problem			function eval.		CPU time		$f(x)$	
name	**n**	**I.P.**	**N**	**A**	**N**	**A**	**N**	**A**
Linear Full Rank	9	x_s	42	42	0.001	0.000	3.600000D+01	3.599990D+01
	9	10 x_s	47	51	0.001	0.001	3.600000D+01	3.599990D+01
Linear Rank 1	7	x_s	159	308	0.003	0.007	8.380200D+00	8.380282D+00
	7	10 x_s	166	246	0.003	0.006	8.380200D+00	8.380282D+00
Linear Rank 1	7	x_s	151	242	0.003	0.005	9.880500D+00	9.880597D+00
(with zero columns and rows)	7	10 x_s	158	247	0.002	0.006	9.880500D+00	9.880597D+00
Rosenbrock	2	x_s	195	158	0.001	0.000	1.382200D-17	7.324709D-09
	2	10 x_s	321	368	0.002	0.002	8.106400D-16	3.569618D-11
Helical valley	3	x_s	124	121	0.001	0.001	2.066500D-13	6.827992D-08
	3	10 x_s	170	157	0.002	0.001	9.369100D-15	1.561482D-07
Powell singular	4	x_s	476	248	0.004	0.002	1.099100D-10	1.050267D-07
	4	10 x_s	571	398	0.005	0.003	9.633300D-11	1.843349D-07
Freudenstein and Roth	2	x_s	76	72	0.000	0.000	4.898420D+01	4.898425D+01
	2	10 x_s	80	162	0.000	0.001	4.898420D+01	4.898425D+01
Bard	3	x_s	113	121	0.001	0.001	8.214800D-03	8.214878D-03
	3	10 x_s	114	145	0.001	0.001	1.774900D+00	8.215013D-03
Kowalik and Osborne	4	x_s	278	220	0.002	0.002	3.075000D-04	3.079336D-04
Meyer	3	x_s	530	438	0.004	0.012	9.762986D+04	9.945488D+04
Watson	6	x_s	937	923	0.014	0.018	2.287600D-03	2.289743D-03
	6	$10x_s$	1717	913	0.027	0.017	2.287600D-03	2.287685D-03
	9	x_s	**8000	1792	0.205	0.083	1.413197D-06	9.782082D-05
	9	$10x_s$	**8000	3566	0.209	0.176	1.420601D-06	5.397333D-06
	12	x_s	**8000	3124	0.335	0.288	4.999653D-08	2.892165D-04
	12	$10x_s$	**8000	**8000	0.322	0.800	5.934833D-09	2.044928D-04
Box 3-dimensional	3	x_s	212	184	0.002	0.002	3.336000D-17	4.188451D-09
Jennrich and Sampson	2	x_s	55	55	0.001	0.000	1.243621D+02	1.243622D+02
Brown and Dennis	4	x_s	191	180	0.002	0.002	8.582220D+04	8.582220D+04
	4	10 x_s	601	602	0.007	0.009	8.582220D+04	8.582220D+04
CheByQuad	6	x_s	177	151	0.002	0.003	4.763800D-13	9.210608D-09
	7	x_s	229	189	0.004	0.004	3.148000D-13	1.204186D-08
	8	x_s	390	202	0.008	0.005	3.516800D-03	3.517317D-03
	9	x_s	529	356	0.013	0.014	2.201700D-13	1.831953D-07
	10	x_s	666	643	0.020	0.032	4.772700D-03	4.772732D-03
	11	x_s	538	408	0.018	0.024	2.799700D-03	2.804779D-03
Brown almost-linear	10	x_s	1255	349	0.039	0.017	1.553300D-12	3.921241D-07
Osborne 1	5	x_s	1012	120	0.014	0.003	6.745600D-05	8.136841D-03
Osborne 2	11	x_s	1709	1024	0.082	0.086	4.013700D-02	4.013847D-02
	11	10 x_s	321	1912	0.017	1.306	1.789000D+00	1.789814D+00
BDQRTIC	8	x_s	432	507	0.009	0.014	1.023890D+01	1.023897D+01
	10	x_s	670	522	0.018	0.025	1.828100D+01	1.828118D+01
	11	x_s	758	614	0.024	0.034	2.226000D+01	2.226062D+01
	12	x_s	781	636	0.027	0.045	2.627200D+01	2.627277D+01
Cube	5	x_s	2842	959	0.027	0.013	7.617000D-07	8.871802D-05
	6	x_s	4625	2123	0.057	0.044	4.772300D-06	1.241336D-05
	8	x_s	6825	2144	0.132	0.067	5.613800D-06	4.786514D-05
Mancino	5	x_s	39	40	0.000	0.001	3.712400D-11	4.006352D-07
	5	$10x_s$	66	49	0.001	0.000	5.742400D-08	4.353584D-05
	8	x_s	52	50	0.002	0.001	1.013600D-08	1.116700D-05
	10	x_s	68	55	0.002	0.002	2.292100D-09	7.324810D-04
	12	x_s	83	66	0.003	0.004	1.216000D-08	3.825228D-04
	12	$10x_s$	131	91	0.006	0.006	1.216000D-08	2.381812D-04
Heart 8	8	x_s	1118	920	0.025	0.031	1.516400D-11	2.043254D-08
	8	$10x_s$	826	616	0.018	0.040	4.906900D+00	4.663999D+00

Table 1: Small scale problems: NEWUOA (N) versus AS-NEW (A)

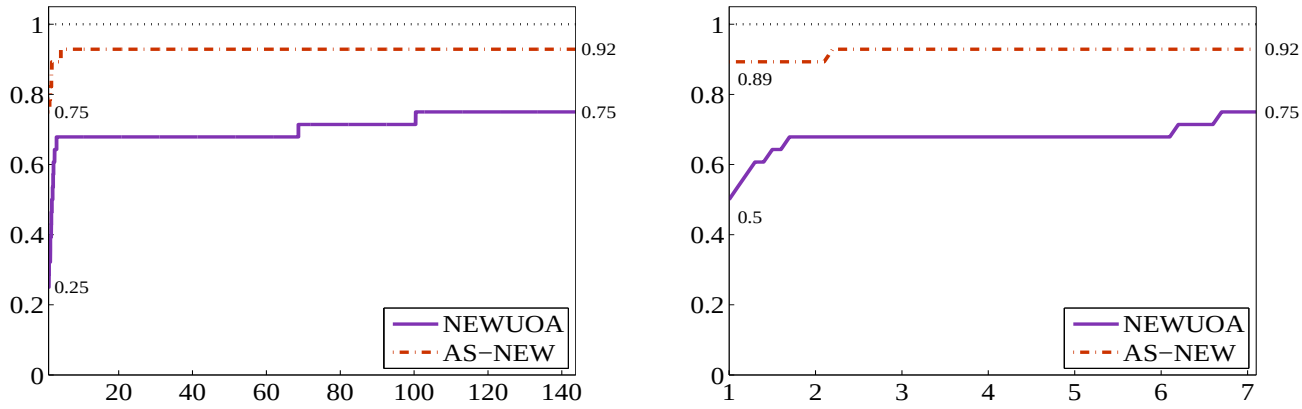

Figure 2: (a) Function evals.: NEWUOA vs. AS-NEW (b) Log-scale

function evaluations in the 75% of the cases. On the other hand, NEWUOA solves successfully the 75% of the problems and perform less function evaluations that AS-NEW only in the 25% of them. Note that in Figure 2 (b) we used a logarithmic scale as it is suggested by Dolan and Moré in [10].

Concerning to the the CPU times NEWUOA was better in most of the cases, but it is well known that for derivative optimization algorithms the goal is reduce the number of function evaluations. Figure 3 shows the comparison profiles for CPU times using medium scale problems.

6 Conclusions

We have presented a new algorithm for solving unconstrained derivative-free optimization problems based on the NEWUOA algorithm of Powell [24], which uses an active set strategy [12] for solving the trust-region subproblems. Since we consider the infinity norm, a box constrained quadratic optimization problem have to be solved in each iteration. Also we use the spectral gradient method [4] to leave a face when an appropriate criterion seems to indicate that it is not worth to continue exploring that face.

We implemented this algorithm and perform numerical experiments using a set of test problem. We compared our algorithm against NEWUOA. The numerical results reported in this paper 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 results and the developments reported by Powell [25] provide new impetus for further development for optimization without derivatives with smooth functions.

Problem		**function eval.**		**CPU time**		$f(x)$	
name	**n**	**N**	**A**	**N**	**A**	**N**	**A**
Arwhead	20	409	409	0.024	0.044	6.827872D-12	4.355508D-07
	40	1441	638	0.280	0.240	6.320278D-12	5.079198D-07
	80	3226	1885	2.730	5.680	8.715428D-11	4.882386D-05
	100	3859	3365	6.170	19.420	4.438583D-11	3.140931D-05
Penalty 1	20	7348	2320	0.520	0.270	1.577771D-04	1.666262D-04
	40	14620	6217	3.180	3.260	3.392511D-04	3.403646D-04
	80	31180	20837	27.330	62.030	7.130502D-04	7.133080D-04
	100	39364	22341	65.950	79.540	9.024910D-04	9.027157D-04
Penalty 2	20	19060	12960	1.380	1.740	6.389751D-03	6.389946D-03
	40	15301	19738	3.400	15.270	5.569125D-01	5.569240D-01
	80	19357	29116	17.130	176.410	1.776315D+03	1.776315D+03
	100	15388	66721	25.004	948.790	9.709608D+04	9.709608D+04
Penalty 3	20	4660	1760	0.057	0.100	3.636060D+02	3.641777D+02
	40	80000	4587	0.410	0.810	1.045973D-03	1.552231D+03
	80	80000	26563	3.520	11.220	6.285525D+03	6.398066D+03
	100	80000	20104	7.720	22.170	9.886938D+03	1.004591D+04
VarDim	20	4791	8509	0.340	1.480	5.365843D-11	8.785758D-07
	40	18725	35543	4.120	38.550	7.200291D-11	2.715300D-07
	80	55635	80000	50.450	585.350	1.943923D-10	6.630367D-04
	100	80000	80000	134.970	1030.730	1.091809D-08	1.281812D-02
Rosenbrock Ext.	20	8585	7774	0.790	1.670	1.491622D-10	1.206377D-03
	40	36435	19054	10.840	18.510	7.109087D-09	3.454462D-03
	80	80000	48315	94.120	202.580	4.092447D-07	2.600383D-04
	100	80000	60547	165.800	438.350	1.960580D-07	4.672712D-04
CheByQuad	20	1817	858	0.140	0.100	4.572955D-03	4.580623D-03
	40	26743	186	8.300	0.041	5.960843D-03	7.193065D-03
	80	73006	727	88.230	1.130	4.931312D-03	5.009640D-03
	100	46964	684	100.350	1.580	8.715769D-03	9.589309D-03

Table 2: Medium scale problems: NEWUOA (N) versus AS-NEW (A)

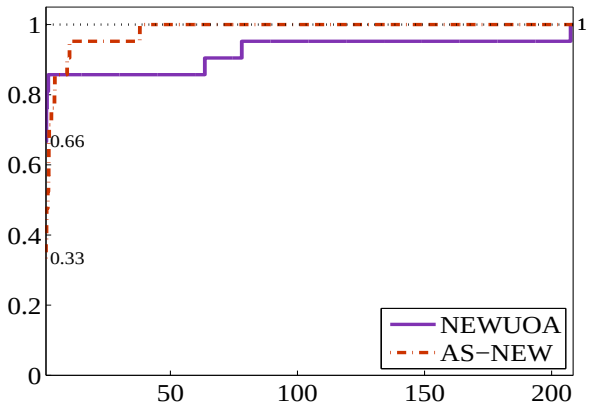

Figure 3: CPU times: NEWUOA vs. AS-NEW

References

[1] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA J. of Numer. Analysis*, V. 8, pp. 141–148, 1988.

[2] F. Berghen and H. Bersini. CONDOR: a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm, *J. Comput. Appl. Math.*, V. 181, pp. 157–175, 2005.

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

[4] E. Birgin, J. M. Martínez and M. Raydán. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10, pp. 1196–1211, 2000.

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

[6] A. Conn, K. Scheinberg and P. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Math. Program.*, V. 79, pp. 397–414, 1997.

[7] 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.

[8] A. Conn, K. Scheinberg and L. Vicente. Introduction to derivative-free optimization. *MPS SIAM Series on Optimization*, SIAM, 2009.

[9] J. Dennis and V. Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, V. 1, N. 4, pp. 448–474, 1991.

[10] E. Dolan and J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, V. 91, pp. 201–213, 2002.

[11] J. Morales, G. Fasano and J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *To appear in Optimization Methods and Software*, V. 24 (1), DOI: 10.1080/10556780802409296. pp. 145–154, 2009.

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

[13] L. Grippo, F. Lampariello and S. Lucidi. Global convergence and stabilization of unconstrained minimization methods without derivatives. *J. Optim. Theory Appl.*, V. 56, pp. 385–406, 1998.

[14] C. Kelley. Iterative methods for Optimization, *SIAM*, Philadelphia, 1999.

[15] T. Kolda, R. Lewis and V. Torczon. Optimization by direct search: Newu perspectives on some classical and modern methods. *SIAM Review*, 45, pp. 385–482, 2003.

[16] M. Marazzi and J. Nocedal, Wedge trust region methods for derivative free optimization. *Math. Program.*, V., 91, pp. 289–305, 2002.

[17] J. Moré and S. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. on Optimization*, V. 20, N. 1, pp. 172–191, 2009.

[18] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, V. 7, pp. 308–313, 1965.

[19] R. Oeuvray. Trust-region methods based on radial basis functions with application to biomedical imaging. *PhD thesis*, Institut de Mathématiques, École Polytechnique Fédérale de Lausanne, Switzerland, 2005.

[20] M. Powell. On the Lagrange functions of quadratic models thar are defined by interpolation. *Optim. Methods Softw.*, V. 16, pp. 289–309, 2001.

[21] M. Powell. UOBYQUA: Unconstrained optimization by quadratic approximation. *Math. Program.*, V., 92, pp. 289–309, 2002.

[22] M. Powell. On trust region methods for unconstrained minimization without derivatives. *Math. Program.*, V. 97, pp. 605–623, 2003.

[23] M. Powell. Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Math. Program.*, V. 100, pp. 183–215, 2004.

[24] M. Powell. The NEWUOA software for unconstrained optimization without dervivatives. *Tech. Report DAMTP 2004/NA08*, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK, 2004.

[25] M. Powell. Developtments of NEWUOA for minimization without dervivatives. *Tech. Report DAMTP 2007/NA05*, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK, 2007.

[26] M. Raydán. On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. of Numerical Analysis*, V. 13, pp. 321–326, 1993.

[27] M. Raydán. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. on Optimization*, V. 7, pp. 26–33, 1997.

[28] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, V. 7, N. 1, pp. 1–25, 1997.

[29] D. Winfield. Functions and functional optimization by interpolation in data tables. *PhD thesis*, Harvard University, Cambrdige, MA, 1969.

[30] D. Winfield. Functions minimization by interpolation in a data set. *J. Inst. Math. Appl.*, V. 12, pp. 339–347, 1973.