SIAM J. OPTIM.
Vol. 7, No. 1, pp. 26–33, February 1997

002

THE BARZILAI AND BORWEIN GRADIENT METHOD FOR THE LARGE SCALE UNCONSTRAINED MINIMIZATION PROBLEM*

MARCOS RAYDAN†

Abstract. The Barzilai and Borwein gradient method for the solution of large scale unconstrained minimization problems is considered. This method requires few storage locations and very inexpensive computations. Furthermore, it does not guarantee descent in the objective function and no line search is required. Recently, the global convergence for the convex quadratic case has been established. However, for the nonquadratic case, the method needs to be incorporated in a globalization scheme. In this work, a nonmonotone line search strategy that guarantees global convergence is combined with the Barzilai and Borwein method. This strategy is based on the nonmonotone line search technique proposed by Grippo, Lampariello, and Lucidi [*SIAM J. Numer. Anal.*, 23 (1986), pp. 707–716]. Numerical results to compare the behavior of this method with recent implementations of the conjugate gradient method are presented. These results indicate that the global Barzilai and Borwein method may allow some significant reduction in the number of line searches and also in the number of gradient evaluations.

Key words. unconstrained optimization, nonmonotone line search, Barzilai and Borwein method, conjugate gradient method

AMS subject classifications. 49M07, 49M10, 90C06, 65K

PII. S1052623494266365

1. Introduction.

In this paper we consider the Barzilai and Borwein gradient method for the large scale unconstrained minimization problem

$$\min_{x \in R^n} f(x), \tag{1}$$

where $f : R^n \to R$. The method is defined by

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k, \tag{2}$$

where g_k is the gradient vector of f at x_k and the scalar α_k is given by

$$\alpha_k = \frac{s_{k-1}^t y_{k-1}}{s_{k-1}^t s_{k-1}}, \tag{3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

Every iteration of the Barzilai and Borwein method requires only $O(n)$ floating point operations and a gradient evaluation. No matrix computations and no line searches are required during the process. The search direction is always the negative gradient direction, but the choice of steplength is not the classical choice of the steepest descent method. In fact, Barzilai and Borwein [1] observed that this new choice of steplength required less computational work and greatly speeded up the convergence of the gradient method for quadratics.

* Received by the editors April 15, 1994; accepted for publication (in revised form) May 16, 1995. This research was supported in part by CDCH-UCV project 03.13.0034.93 and by NSF grant CHE-9301120.

http://www.siam.org/journals/siopt/7-1/26636.html

† Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (mraydan@conicit.ve).

More interesting from a theoretical point of view is that the method does not guarantee descent in the objective function. Barzilai and Borwein [1] presented a convergence analysis in the two-dimensional quadratic case. They established, for that particular case, R-superlinear convergence. However, Fletcher [5] argued that, in general, only R-linear convergence should be expected. Later, Raydan [18] established global convergence for the strictly convex quadratic case with any number of variables. This result has been recently extended to the (not necessarily strictly) convex quadratic case by Friedlander, Martinez, and Raydan [6] to incorporate the method in a box constrained optimization technique.

Glunt, Hayden, and Raydan [9] established a relationship with the shifted power method that adds understanding to the significant improvement obtained with the choice of steplength given by (3). In particular, they applied the Barzilai and Borwein gradient method to find local minimizers of nonquadratic functions that appear in the determination of molecular structures from nuclear magnetic resonance data. For this application, it was possible to choose good starting values and convergence was observed. However, in general, for the nonquadratic case the method needs to be incorporated in a globalization scheme.

The object of this work is to embed the Barzilai and Borwein gradient method in a globalization strategy that accepts the steplength given by (3) as frequently as possible and that only requires storage of first-order information during the process.

This paper is organized as follows. In section 2 we present a globalization strategy suitable for the Barzilai and Borwein method. This strategy is based on the nonmonotone line search technique of Grippo, Lampariello, and Lucidi [10] for Newton's method. We discuss the properties of this new algorithm and establish global convergence under mild assumptions. In section 3 we present some preliminary numerical results to compare the behavior of our global new method with recent implementations of the conjugate gradient method for the nonquadratic case. Finally, in section 4 we present some concluding remarks.

2. Globalization strategy. Standard methods for the solution of (1) usually generate a sequence of iterates for which a sufficient decrease in the objective function f is enforced at every iteration. In many cases, the global strategy consists of accepting the steplength, in the search direction, if it satisfies the well-known Armijo–Goldstein–Wolfe conditions. Practical line search schemes have been developed to enforce these conditions when combined with Newton, quasi-Newton, and conjugate gradient methods. For a complete discussion on this topic see [3], [4], and [14].

There are some disadvantages to forcing the Armijo–Goldstein–Wolfe conditions when combined with the Barzilai and Borwein gradient method. One of the disadvantages is that forcing decrease at every iteration will destroy some of the local properties of the method. As it is argued in Fletcher [5] and Glunt, Hayden, and Raydan [9], the choice of steplength (3) is related to the eigenvalues of the Hessian at the minimizer and not to the function value. Moreover, since the search direction is always the negative gradient direction, forcing decrease at every iteration will reduce the method to the steepest descent method, which is known for being slow.

Therefore, we will enforce a much weaker condition of the form

$$f(x_{k+1}) \leq \max_{0\leq j\leq M} f(x_{k-j}) + \gamma g_k^t (x_{k+1} - x_k), \tag{4}$$

where M is a nonnegative integer and γ is a small positive number. This type of condition (4) was introduced by Grippo, Lampariello, and Lucidi [10] and successfully applied to Newton's method for a set of test functions. Recently, the same type

of nonmonotone line search technique has been incorporated into a variety of optimization algorithms. See, for instance, [11], [15], and [16]. Condition (4) allows the objective function to increase at some iterations and still guarantees global convergence. This feature fits nicely with the nonmonotone behavior of the Barzilai and Borwein gradient method. We now present the proposed algorithm.

GLOBAL BARZILAI AND BORWEIN (GBB) ALGORITHM.

Given x_0, α_0, integer $M \geq 0$, $\gamma \in (0,1)$, $\delta > 0$,
$0 < \sigma_1 < \sigma_2 < 1$, $\quad 0 < \epsilon < 1$. Set $k = 0$.

Step 1: If $\|g_k\|$=0 stop

Step 2: If $\alpha_k \leq \epsilon$ or $\alpha_k \geq 1/\epsilon$ then set $\alpha_k = \delta$

Step 3: Set $\lambda = 1/\alpha_k$

Step 4: (nonmonotone line search)
If $f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k,M)}(f_{k-j}) - \gamma \lambda g_k^t g_k$
then set $\lambda_k = \lambda$, $x_{k+1} = x_k - \lambda_k g_k$, and go to Step 6

Step 5: Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\lambda = \sigma\lambda$, and go to Step 4

Step 6: Set $\alpha_{k+1} = -(g_k^t y_k)/(\lambda_k g_k^t g_k)$, $k = k + 1$, and go to Step 1.

Remarks. (1) The object of Step 2 is to avoid uphill directions and to keep the sequence $\{\lambda_k\}$ uniformly bounded. In fact, for all k

$$0 < \min\left(\epsilon, \frac{1}{\delta}\right) \leq \lambda_k \leq \max\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right).$$

(2) The GBB algorithm cannot cycle indefinitely between Steps 4 and 5. Indeed, since $\lambda g_k^t g_k > 0$ and $\gamma < 1$, for sufficiently small values of λ the condition in Step 4 is satisfied.

(3) Since $s_k = -\lambda_k g_k$, then the definition of α_{k+1} given in Step 6 is equivalent to the one given in (3). The advantage of the definition used in the algorithm is that it avoids the storage of the vector s_k and reduces to $3n$ locations the storage requirements of the GBB algorithm.

(4) For $k = 0$ the condition in Step 4 reduces to the Armijo α condition. For $k > 0$ the objective function might increase at some iterations. However, $f(x_k) \leq f(x_0)$ for all k, and so the level set $\{x : f(x) \leq f(x_0)\}$ contains the entire sequence of iterates $\{x_k\}$.

The convergence properties of the GBB algorithm are stated in the following theorem.

THEOREM 2.1. *Assume that $\Omega_0 = \{x : f(x) \leq f(x_0)\}$ is a bounded set. Let $f : R^n \to R$ be continuously differentiable in some neighborhood N of Ω_0. Let $\{x_k\}$ be the sequence generated by the GBB algorithm. Then either $g(x_j) = 0$ for some finite j, or the following properties hold:*

(i) $\lim_{k\to\infty} \|g_k\| = 0$;

(ii) *no limit point of $\{x_k\}$ is a local maximum of f;*

(iii) *if the number of stationary points of f in Ω_0 is finite, then the sequence $\{x_k\}$ converges.*

Proof. In order to establish (i), we make use of the first part of the proof of the convergence theorem in [10, p. 709].

Let us define $m(k) = \min(k, M)$. Clearly, $m(0) = 0$ and

$$0 \le m(k) \le \min(m(k-1) + 1, M) \quad \text{for } k \ge 1.$$

Moreover, there exists a positive constant a such that $0 < \lambda_k \le a$ for all k. Indeed, in the GBB algorithm $a = \max(\epsilon^{-1}, \delta^{-1})$. Finally, there exist positive numbers c_1 and c_2 such that the search direction d_k satisfies $g_k^t d_k \le -c_1 \|g_k\|_2^2$ and $\|d_k\|_2 \le c_2 \|g_k\|_2$. In fact, in the GBB algorithm, the search direction d_k is $-g_k$ for all k and so $c_1 = c_2 = 1$. Therefore, we obtain equation (14) in [10, p. 711] that in our case reduces to

$$\lim_{k\to\infty} \lambda_k \|g_k\|_2 = 0.$$

Since $\lambda_k \ge \min(\epsilon, \frac{1}{\delta})$ for all k, then part (i) holds. Assertions (ii) and (iii) follow directly from the convergence theorem in [10]. □

Notice that, forcing the weak condition (4), the sequence $\{x_k\}$ generated by the GBB algorithm has the following property:

$$\lim_{k\to\infty} \|g_k\| = 0.$$

This is in sharp contrast to the conjugate gradient methods (Fletcher–Reeves, Polak–Ribière, etc.) for which much stronger conditions have to be imposed to obtain the weaker result:

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

For a further discussion on the convergence properties of the conjugate gradient methods see Nocedal [14] and Gilbert and Nocedal [8].

3. Numerical results. In this section we present numerical results to compare the behavior of the GBB algorithm with two different implementations of the conjugate gradient method for the nonquadratic case. In particular, we compare the GBB algorithm with the well-known routine CONMIN of Shanno and Phua [19], which includes automatic restarts and requires $7n$ storage locations. We also compare the GBB algorithm with the Polak–Ribière implementation of Gilbert and Nocedal [8] (PR^+) that requires $4n$ storage locations and for which global convergence was established under mild assumptions. The line search for the PR^+ method is based on the algorithm of Moré and Thuente [13] and is fully described in [8]. It is worth mentioning that for the classical Polak–Ribière method no satisfactory global convergence result has been found and a negative convergence result has been established; see Powell [17]. For this lack of theoretical support we have decided to compare the new algorithm with PR^+ instead of the classical Polak–Ribière method, although they behave similarly in practice.

The problems used in our tests include well-known large functions and two new strictly convex functions. Table 1 lists the problems and the references for descriptions of the test functions and the starting points. For the problems of Moré, Garbow, and Hillstrom [12] we use the standard starting vector. In this paper, we only describe the new functions:

Strictly Convex 1:

$$f(x) = \sum_{i=1}^{n} (e^{x_i} - x_i); \quad x_0 = \left(\frac{1}{n}, \ldots, \frac{i}{n}, \ldots, 1\right)^t.$$

TABLE 1
Test problems.

Problem	Name	Reference
1	Strictly Convex 1	
2	Strictly Convex 2	
3	Brown almost linear	Moré et al. [12]
4	Trigonometric	Moré et al. [12]
5	Broyden tridiagonal	Moré et al. [12]
6	Oren's power	Garg and Tapia [7]
7	Extended Rosenbrock	Moré et al. [12]
8	Penalty 1	Moré et al. [12]
9	Tridiagonal 1	Buckley and LeNir [2]
10	Variably dimensioned	Moré et al. [12]
11	Extended Powell	Moré et al. [12]
12	Generalized Rosenbrock	Moré et al. [12]
13	Extended ENGLV1	Toint [20]
14	Extended Freudenstein and Roth	Toint [20]
15	Wrong Extended Wood	Toint [20]

Strictly Convex 2:

$$f(x) = \sum_{i=1}^{n} \frac{i}{10}(e^{x_i} - x_i); \quad x_0 = (1, 1, \ldots, 1)^t.$$

Clearly, the unique minimizer of Strictly Convex 1 and Strictly Convex 2 is given by $x_\star = (0, \ldots, 0)^t$. The Hessian of Strictly Convex 1 at $x_\star$ is the identity matrix, and the Hessian of Strictly Convex 2 at $x_\star$ has n distinct eigenvalues.

All the experiments were run on a SparcStation 1 in double precision FORTRAN with a machine epsilon of about 2×10^{-16}. For the GBB algorithm we used $\gamma = 10^{-4}$, $\epsilon = 10^{-10}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\alpha_0 = 1$, and $M = 10$. We have chosen the parameter ϵ to be a very small number in order to accept the Barzilai and Borwein step as many times as possible. However, if the condition in Step 2 was satisfied at iteration k, then the parameter δ was chosen in the following way:

$$\delta = \begin{cases} 1 & \text{if} \quad \|g_k\|_2 > 1, \\ \|g_k\|_2^{-1} & \text{if} \quad 10^{-5} \le \|g_k\|_2 \le 1, \\ 10^5 & \text{if} \quad \|g_k\|_2 < 10^{-5}. \end{cases}$$

Notice that, with this choice of δ, the sequence $\{\lambda_k\}$ remains uniformly bounded. In Step 5, σ is chosen by means of a quadratic interpolation described in [3, p. 127]. All runs were stopped when

$$\|g_k\|_2 \le 10^{-6}(1 + |f(x_k)|),$$

and we verified that the three methods converged to the same solution point.

The numerical results are shown in Table 2. We report number of iterations (IT), CPU time in seconds (Time), number of function evaluations (f), number of gradient evaluations (g), and number of line searches (LS) required by the GBB method during the process, i.e., number of iterations for which the GBB algorithm goes through Step 5 at least once. Every time GBB requires a line search, it needs additional function evaluations and no additional gradient evaluations. On the other hand, CONMIN and PR^+ require a line search at every iteration and as many function

TABLE 2
Results for GBB, CONMIN, and PR^+.

	GBB					CONMIN			PR^+		
P(n)	IT	f	g	LS	Time	IT	f-g	Time	IT	f-g	Time
1(100)	8	8	8	0	0.04	15	38	0.16	6	17	0.14
1(1000)	8	8	8	0	0.28	15	38	1.23	7	22	0.92
1(10000)	8	8	8	0	2.8	15	38	12.57	7	22	8.45
2(100)	52	57	52	4	0.29	40	81	0.55	33	69	0.45
2(500)	74	80	74	5	1.61	63	127	4.1	44	92	2.34
2(1000)	82	91	82	7	3.49	71	145	9.13	40	84	4.17
3(100)	3	3	3	0	0.01	3	7	0.03	2	4	0.03
3(1000)	4	4	4	0	0.1	15	38	0.94	F	F	F
3(10000)	57	72	57	10	25.5	17	41	22.3	F	F	F
4(100)	76	81	76	4	0.86	51	108	1.33	54	121	1.1
4(1000)	93	106	93	13	9.6	53	112	13.1	58	132	10.1
4(10000)	89	99	89	10	83.3	59	126	134.	61	133	97.
5(100)	34	34	34	0	0.13	33	67	0.34	31	70	0.42
5(1000)	40	40	40	0	1.1	38	75	3.8	32	75	3.47
5(3000)	44	45	44	1	3.7	35	71	10.8	31	71	9.82
6(100)	105	112	105	7	0.32	49	99	0.3	39	87	0.25
6(1000)	310	378	310	54	6.8	158	320	10.8	114	236	6.42
6(10000)	1351	1750	1351	263	325.	464	937	365.	355	719	207.
7(100)	69	91	69	15	0.22	19	47	0.12	25	73	0.24
7(1000)	93	118	93	20	1.73	30	73	1.92	23	70	1.45
7(10000)	70	92	70	11	14.3	28	69	16.3	20	64	13.9
8(100)	48	49	48	1	0.16	27	65	0.23	53	204	0.75
8(1000)	57	57	57	0	1.22	25	55	1.44	40	164	4.15
8(10000)	62	62	62	0	14.2	25	55	15.	40	164	43.2
9(100)	167	191	167	18	0.55	80	161	0.8	78	158	0.7
9(1000)	878	1152	878	186	21.3	306	613	25.8	295	593	18.3
10(100)	38	38	38	0	0.13	13	29	0.1	7	39	0.16
10(1000)	54	54	54	0	1.22	27	62	1.54	F	F	F
11(100)	740	988	740	136	3.5	47	95	0.48	190	434	1.6
11(1000)	815	1125	815	163	32.6	43	87	4.1	99	238	10.6
12(100)	1429	1869	1429	342	4.85	254	516	2.3	258	533	2.63
12(500)	4452	5622	4452	1087	51.	1082	2280	45.3	1072	2162	39.7
13(100)	26	26	26	0	0.1	13	27	0.15	17	43	0.21
13(1000)	23	23	23	0	0.54	12	25	0.81	13	45	1.32
13(10000)	21	21	21	0	5.16	11	23	7.3	9	32	9.3
14(100)	438	560	438	102	2.0	13	27	0.14	14	39	0.3
14(1000)	288	377	288	69	10.3	12	25	1.13	19	50	2.4
14(10000)	119	151	119	21	44.2	11	23	11.1	8	30	16.8
15(100)	76	85	76	8	0.3	25	53	0.25	54	127	0.62
15(1000)	80	87	80	5	2.32	34	70	2.83	29	66	2.1

evaluations as gradient evaluations during the process. Hence, for CONMIN and PR^+, we report function and gradient evaluations under the label (f-g). The letter F that appears under the multicolumn PR^+ means that the run was stopped because the line search procedure failed to find a steplength. In those cases, we were not able to report any information for the PR^+ method. The results of Table 2 are summarized in Table 3. We report in Table 3 the number of problems for which each method was a winner in number of iterations, number of gradient evaluations, and CPU time.

We observe that the GBB method out performs CONMIN and PR^+ in number of gradient evaluations and CPU time, except for problems with a very ill-conditioned Hessian at the solution. For some of these problems, GBB is still competitive in

TABLE 3
Number of problems for which a method was a winner.

Method	IT	g	Time
GBB	1	19	22
CONMIN	17	12	10
PR^+	22	9	8

CPU time. However, if the Hessian is singular at the solution as in problem 11, then CONMIN and PR^+ clearly out perform GBB.

CONMIN and PR^+ out perform GBB in number of iterations, except for problems with a well-conditioned Hessian at the solution, in which case the number of iterations is quite similar. In some of those cases (problems 1 and 5), the difference in computing time is remarkable.

4. Concluding remarks. The Barzilai and Borwein method can be incorporated in a globalization strategy that preserves the good features of the method and only requires $3n$ storage locations. Since the search direction is always the negative gradient direction, it is trivial to ensure that descent directions are generated at every iteration. This is in sharp contrast to the conjugate gradient methods, for which a very accurate line search has to be performed at every iteration to generate descent directions.

Our numerical experiments seem to indicate that the global Barzilai and Borwein algorithm is competitive and sometimes preferable to recent and well-known implementations of the conjugate gradient method. However, further numerical investigation needs to be done to establish this conclusively.

We observe that the GBB algorithm requires few line searches. In the worst case (problem 12), it requires 1 line search out of every 5 iterations. Moreover, we have observed that near the solution the GBB method does not require any line search. At this point, we would like to stress that no local convergence analysis has been presented to support this observation. All we can say, so far, to explain the local behavior of the GBB method is that the Barzilai and Borwein method, given by (2) and (3), is globally convergent for convex quadratic functions.

Finally, we would like to comment on the choice of the parameter M in the GBB algorithm. We have tested the same set of problems with different values of M ranging from 5 to 20. In general, we observed similar results to the ones presented in Tables 2 and 3, except for problems with a singular or very ill-conditioned Hessian at the solution. For these problems, the behavior of the method is very sensitive to the choice of M. For example, using $M = 20$ in problem 11 with $n = 1000$, convergence is obtained after 365 iterations, 36 line searches, 451 function evaluations, and 12.8 seconds of execution time. These results represent a significant improvement over the ones reported in Table 2 with $M = 10$. On the other hand, using $M = 5$ the results obtained are worse than the ones in Table 2. Therefore, for the singular or very ill-conditioned case, the choice of the parameter M is a delicate issue and merits further investigation.

Acknowledgments. The author thanks Jorge Nocedal for providing all the required routines to test the PR^+ method. He also thanks Jose Mario Martinez and two anonymous referees for their constructive suggestions.

REFERENCES

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

[2] A. Buckley and A. LeNir, *QN-like variable storage conjugate gradients*, Math. Programming, 27 (1983), pp. 155–175.

[3] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[4] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, 1987.

[5] R. Fletcher, *Low storage methods for unconstrained optimization*, in Lectures in Applied Mathematics, Vol. 26, American Mathematical Society, Providence, RI, 1990, pp. 165–179.

[6] A. Friedlander, J. M. Martinez, and M. Raydan, *A new method for large-scale box constrained convex quadratic minimization problems*, Optim. Methods and Software, 5 (1995), pp. 57–74.

[7] N. K. Garg and R. A. Tapia, *QDN: A Variable Storage Algorithm for Unconstrained Optimization*, Technical report, Department of Mathematical Sciences, Rice University, Houston, TX, 1977.

[8] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21–42.

[9] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114-120.

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

[11] L. Grippo, F. Lampariello, and S. Lucidi, *A class of nonmonotone stabilization methods in unconstrained optimization*, Numer. Math., 59 (1991), pp. 779–805.

[12] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17–41.

[13] J. J. Moré and D. J. Thuente, *On line search algorithms with guaranteed sufficient decrease*, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1990, preprint MCS-P153-0590.

[14] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numerica, 1 (1992), pp. 199–242.

[15] J. S. Pang, S. P. Han, and N. Rangaraj, *Minimization of locally lipschitzian functions*, SIAM J. Optim., 1 (1991), pp. 57–82.

[16] E. R. Panier and A. L. Tits, *Avoiding the Maratos effect by means of a nonmonotone line search* I. *General constrained problems*, SIAM J. Numer. Anal., 28 (1991), pp. 1183–1195.

[17] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, in Lecture Notes in Mathematics, Vol. 1066, Springer-Verlag, Berlin, 1984, pp. 122–141.

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

[19] D. F. Shanno and K. H. Phua, *Remark on algorithm 500: Minimization of unconstrained multivariate functions*, ACM Trans. Math. Software, 6 (1980), pp. 618–622.

[20] Ph. L. Toint *Test Problems for Partially Separable Optimization and Results for the Routine PSPMIN*, Report Nr 83/4, Department of Mathematics, Facultés Universitaires de Namur, Namur, Belgium, 1983.