Accelerated Modified Memory-less SR1 Method with Cubic Regularization for Unconstrained Optimization

Neculai Andrei
Center for Advanced Modeling and Optimization,
Academy of Romanian Scientists,
54, Splaiul Independenţei, Sector 5,
Bucharest, ROMANIA
E-mail: neculaiandrei70@gmail.com

Technical Report 6/2021
April 28, 2021

Abstract. Symmetric rank-one updates (SR1) are known to be quasi-Newton algorithms, which are enough fast and often provide better approximations to the true Hessian of the minimizing function than the most appreciated rank-two approaches. However, these properties are guaranteed only in some conditions which frequently do not hold. The main deficiency of SR1 update is that it does not guarantee the positive definiteness for the Hessian estimate, thus ruining the convergence of the corresponding algorithm. This paper introduces the memory-less SR1 method with cubic regularization as a modification of the classical SR1 update for which the updating term is imposed to be positive definite along the iterations. In case this is not possible, then the algorithm reduces to the scaled memory-less SR1 method. The global convergence is proved under mild assumptions. Intensive numerical experiments on a collection of 800 unconstrained optimization problems with the number of variables in the range [1000, 10000], shows the efficiency and robustness of this approach over the scaled memory-less SR1 method and over the BFGS from CONMIN.

Key words: Unconstrained optimization, SR1 method, Cubic regularization, Modified secant equation, Memory-less SR1, Modified memory-less SR1

Mathematics Subject Classification: 90C30, 90C53, 90-08

1 Introduction

Many algorithms for solving the unconstrained optimization problem $\min f(x)$, where $x \in \mathbb{R}^n$ and $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded from below employ a quadratic model of the function f. The Newton's method and the quasi-Newton methods use a second-order Taylor series of the minimizing function with either

an explicit or an approximated Hessian matrix. The quasi-Newton methods use an approximation to the Hessian, or an approximation to the inverse Hessian, which are updated at each step of the optimization algorithm. These algorithms are efficient and robust for minimizing functions that satisfy certain assumptions and have a super-linear rate of local convergence. In the frame of quasi-Newton methods many variants of the updating formula for the approximation to the Hessian (or to the inverse Hessian) were suggested: *symmetric rank-one* (SR1) [Broyden, 1967; Davidon, 1959, 1968; Fiacco and McCormick, 1968; Wolfe, 1969, 1971] and *the rank-two* such as the Davidon-Fletcher-Powell (DFP) update [Davidon, 1991; Fletcher and Powell, 1963], Powell-symmetric-Broyden (PSB) [Powell, 1970] and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update [Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970]. A unifying framework for many of these updates, including both rank-one and rank-two updates was given by Huang [1970].

Theoretical studies and intensive numerical experiments proved that among the quasi-Newton methods the BFGS is the most effective. However on certain problems BFGS method may require a large number of iterations and function and gradient evaluations. The sources of its inefficiency may be caused by a poor initial approximation to the Hessian, or more importantly, by the ill-conditioning of the Hessian approximations along the iterations. To improve the efficiency and the robustness of the BFGS and to overcome the difficulties, some modified versions of it were given. All these modified BFGS methods can be classified into four classes: the scaling of the BFGS update matrix, the memory-less BFGS method, the BFGS update with modified secant equation and the modified BFGS method using different line search conditions for stepsize computation. Intensive numerical experiments on minimizing functions with different number of variables and complexities showed that all the modified BFGS methods are more efficient and more robust than its unmodified version [Contreras and Tapia, 1993; Oren and Luenberger, 1974; Oren and Spedicato, 1976; Yabe, Martinez and Tapia, 2004; Biggs, 1971, 1973; Liao, 1997; Nocedal and Yuan, 1993; Andrei, 2018a, 2018b, 2018c, 2020; Yuan, 1991; Yuan and Byrd, 1995; Al-Baali, 1998; Arzam, Babaie-Kafaki and Ghanbari, 2017; Yuan, Sheng, Wang, Hu and Li, 2018; Dehmiry, 2020].

On the other hand, quasi-Newton methods based on the symmetric rank-one (SR1) update are known to be fast and often under certain conditions, which frequently are not satisfied, provide better approximations to the true Hessian of the minimizing function than the rank-two approaches. Additionally, SR1 method is affected by the lack of guarantee of positive definiteness for the Hessian approximation, thus ruining the performances of the corresponding algorithms. The convergence of this method was studied *inter alia* by Conn, Gould and Toint (1991), Khalfan, Byrd and Schnabel (1993). In order to remedy the deficiencies of the SR1 method, to relax the conditions on the proof of convergence for improving the speed and the accuracy of the solution and to get at every step of the algorithm positive definite approximations to the Hessian, Benson and Shanno (2018) introduced cubic regularization in SR1 method.

Even not new, recently we notice a great interest using cubic regularization to the quadratic model of the minimizing function when computing the search direction within a

nonlinear programming algorithm. The idea of using the cubic regularization into the context of the Newton method first appeared in Griewank (1981) and was later developed by many authors, proving its convergence and complexity (Nesterov and Polyak, 2006), (Cartis, Gould and Toint, 2011a, 2011b), (Gould, Porcelli and Toint, 2012), (Bianconcini, Liuzzi, Morini and Sciandrone, 2013), (Bellavia and Morini, 2014), Griewank, Fisher and Bosse, (2014), (Bianconcini and Sciandrone, 2016), (Hsia, Sheu and Yuan, 2017). For a minimizing function f its p-regularization model is constructed by adding a p-th regularization term to the quadratic estimation of f. The idea is to construct and minimize a local quadratic approximation of the minimizing function with a weighted regularization term $(\sigma_k / p)\|x\|^p$, $p > 2$. The most common choice to regularize the quadratic approximation is the p-regularization with $p = 3$, which is known as the *cubic regularization*. Griewank proved that any accumulation point of the sequence generated by minimizing the p-regularized subproblem is a second-order critical point of f, i.e., a point $\bar{x} \in \mathbb{R}^n$ satisfying $\nabla f(\bar{x}) = 0$ and $\nabla^2 f(\bar{x})$ semipositive definite. Later, Nesterov and Polyak (2006) proved that the cubic regularization method has a better global iteration complexity bound than the one for the steepest descent method. Based on these results, Cartis, Gould and Toint (2011a, 2011b, 2012) proposed the Adaptive Regularization algorithm using Cubics (ARC) for minimizing the function f, where the sequence of the regularization parameter $\{\sigma_k\}$ is dynamically determined and the p-regularized subproblems are inexactly solved. In ARC the regularization parameter is initialized at the beginning of the iterations and then at each iteration it is updated using different schemes based on sufficient descent. The performances of the resulting algorithm are strongly dependent by the updating scheme of the regularization parameter. Andrei (2020) and Cartis, Gould and Toint (2011a) suggested updating procedures based on the trust-region ratio. Another procedure using an interpolation condition was given by Zhao, Liu and Liu (2020). Benson and Shanno (2014) proposed a hybrid approach that uses cubic regularization within the Newton method only during the iterations in which the Hessian is indefinite. In this approach computing the cubic step is equivalent to solving a linear system for certain value of the Levenberg-Marquardt perturbation parameter. A similar hybrid approach, but in the context of the SR1 quasi-Newton method with line search was applied by Benson and Shanno (2018). They introduced the hybrid cubic approach for SR1 in which the original SR1 update is used when the search direction is descent and a modified update otherwise.

Having in view the above mentioned developments of the BFGS method as well as the cubic regularization it is quite natural to try to extend them to the SR1 method. In this paper we present the memory-less SR1 method with cubic regularization which combines both the memory-less and cubic techniques. Generally, by memory-less techniques we mean that the approximation to the Hessian in context of quasi-Newton methods, at a given iteration, is computed by updating the identity matrix. Cubic regularization is introduced to improve the convergence for both the speed and the accuracy and to provide at every iteration a positive definite approximation to the Hessian. Besides, cubic regularization tries to compensate the loss of information due to the updating the identity matrix and not the approximation to the Hessian from the previous iteration. Andrei (2021) developed an accelerated scaled memory-less SR1 method in which at every

iteration the identity matrix is updated by SR1 formula and the updating term is scaled with a parameter determined by the sufficient descent or by the conjugacy conditions. This paper introduces the modified memory-less SR1 method with cubic regularization which is obtained by the modified secant equation corresponding to a cubic over-estimator of the minimizing function.

The structure of the paper is as follows. Section 2 is dedicated to present the quasi-Newton SR1 update and the general quasi-Newton algorithm. The main drawbacks of the SR1 update are discussed. The scaled memory-less quasi-Newton SR1 update is developed in Section 3. The scaling parameter is determined in such a way to ensure that the search direction is a descent well defined one. Two procedures for computing the scaling parameter are described. The first one is based on conjugacy condition, while the second one is based on the sufficient descent condition. In Section 4 the SR1 with cubic regularization is presented. Both the modified SR1 update formula and the modified memory-less SR1 update formula are detailed. The convergence of the algorithm is proved in Section 5. Under mild conditions it is proved that the modified memory-less SR1 update matrix with cubic regularization is bounded above in norm and the modified memory-less SR1 search direction is a descent direction, Section 6 presents the numerical results of the accelerated algorithms described in this paper on a collection of 800 unconstrained optimization test problems with the number of variables in the range [1000, 10000]. The acceleration of the algorithms is taken in sense of Andrei (2009). It is proved that the accelerated modified memory-less SR1 method with cubic regularization is more efficient and more robust than the accelerated scaled memory-less SR1 method.

2 The Quasi-Newton SR1 Update

Starting from an initial point x_0 a line search algorithm for solving the unconstrained optimization problem

$$\min f(x), \tag{1}$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded from below, determine a descent direction d_k and a stepsize α_k on this direction and compute the next approximation of the minimum point as

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0,1,\ldots. \tag{2}$$

Often the stepsize α_k is computed as solution of the Wolfe line search conditions

$$f(x_k + \alpha_k d_k) < f(x_k) + \rho \alpha_k g_k^T d_k, \tag{3.a}$$
$$\nabla f(x_k + \alpha_k d_k)^T d_k > \sigma \nabla f(x_k)^T d_k. \tag{3.b}$$

where σ and ρ are scalar parameters with $0 < \rho < \sigma < 1$.

The structure of a quasi-Newton method with Wolfe line search, in the variant in which an approximation to the Hessian is used, can be presented as follows

General quasi-Newton Algorithm

1) Set $k = 0$. Consider an initial point x_0 and an initial Hessian approximation B_0, Select some values for the Wolfe line search conditions σ and ρ with $0 < \rho < \sigma < 1$. Select a sufficiently small parameter $\varepsilon > 0$ used in the criterion for stopping the iterations.
2) **While** $\|\nabla f(x_k)\| > \varepsilon$ do
   a. Evaluate the gradient $g_k = \nabla f(x_k)$.
   b. Solve the system $B_k d_k = -g_k$ subject to the search direction d_k
   c. Find the stepsize $\alpha_k > 0$ which satisfies the standard Wolfe line search conditions (3.a) and (3.b)
   d. Compute: $x_{k+1} = x_k + \alpha_k d_k$ and set $k = k + 1$.
   e. Update B_{k+1}, according the quasi-Newton formula in use.
3) **End while**.

Another variant of the above general quasi-Newton algorithm, often used in the current implementations of the quasi-Newton algorithms for solving the problem (1), at every iteration uses an approximation to the inverse Hessian H_k. In this variant in step 1) of the algorithm instead of B_0 an initial approximation to the inverse Hessian H_0 is selected and in step 2.b) instead of solving the linear algebraic system $B_k d_k = -g_k$, the search direction is computed as $d_k = -H_k g_k$. In step 2.e) instead of updating the approximation to the Hessian B_{k+1}, a formula for approximation to the inverse Hessian H_{k+1} is used. The quasi-Newton algorithms differentiate in step 2.e) concerning the updating the approximation to the Hessian B_{k+1}, or to the inverse Hessian H_{k+1}.

In the quasi-Newton methods the basic requirement for the updating formula to the Hessian is the so called the secant equation to be satisfied at each iteration, namely

$$B_{k+1} s_k = y_k \text{ or } H_{k+1} y_k = s_k, \tag{4}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

The symmetric rank-one SR1 update formula, in which we are interested in this paper, can be derived as solution of the following simple problem. "*Given a symmetric matrix B_k and the vectors s_k and y_k, finds a new symmetric matrix B_{k+1} such that $B_{k+1} - B_k$ has rank one and such that the secant equation $B_{k+1} s_k = y_k$ is satisfied*." It is easy to see that if $(y_k - B_k s_k)^T s_k \neq 0$, then the unique solution of the above problem is

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{5}$$

If $y_k = B_k s_k$ then the solution is $B_{k+1} = B_k$. However, if $(y_k - B_k s_k)^T s_k = 0$ and $y_k \neq B_k s_k$, then there is no solution to the problem. Therefore, to get the SR1 method in step 2.e) of

the general quasi-Newton algorithm the update B_{k+1} approximation to the Hessian is computed like in (5).
Let H_k be the inverse approximation to the Hessian at iteration k. By using the Sherman-Morrison-Woodbury formula in (5), the following update to the inverse Hessian for SR1 is obtained

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{6}$$

This variant of the algorithm is only applicable in cases in which the inverse H_k exists.

Now a comparison between the BFGS and SR1 updates is welcomed. As we know the most effective quasi-Newton updating of the approximations to the Hessian is considered the BFGS formula [Nocedal, 1992], [Nocedal & Wright, 2006] where

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{7}$$

which is a rank-two update that satisfies the secant equation (4). If H_k is the inverse approximation to the Hessian at iteration k, then by applying the Sherman-Morrison-Woodbury formula twice the following update to the inverse Hessian for BFGS is

$$H_{k+1} = H_k - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{y_k^T s_k} + \left(1 + \frac{y_k^T H_k y_k}{y_k^T s_k}\right)\frac{s_k s_k^T}{y_k^T s_k}. \tag{8}$$

The most important properties of BFGS are as follows. If H_k is positive definite, then also H_{k+1} given by (8) is positive definite for any k, provided that $y_k^T s_k > 0$ (which always is satisfied when the Wolfe line search (3) are satisfied). Therefore, if H_0 is chosen to be positive definite, then the rest of all the approximations H_k will also be positive definite. Also BFGS has the self-correcting property, i.e. if H_k incorrectly approximates the curvature of the minimizing function and this estimate slows down the iteration, then the inverse Hessian approximation will tend to correct itself in the next few iterations. The self-correcting property depends on the quality of the implementation of the Wolfe line search. For the Wolfe line search, always the initial value $\alpha = 1$ is tried and this produce superlinear convergence of the method. All these properties of BFGS update make this quasi-Newton method one of the best in this class. However, the things are not what they seem to be.

There are great differences between the SR1 update (5) and the BFGS update (7). Firstly, observe that SR1 is a rank-one update of the Hessian and BFGS is a rank-two update, both of them satisfying the secant equation. Secondly, there are some drawbacks of SR1 which are not encountered in BFGS. The main drawbacks of SR1 update are as follows.

1) The denominator $(y_k - B_k s_k)^T s_k$ of the SR1 update term in (5) may vanish, i.e. $(y_k - B_k s_k)^T s_k \cong 0$, cases in which B_{k+1} is not well-defined.

2) The step directions computed by using the SR1 updating formula given by (5) may no longer be uniform linear independent, thus leading to slow down the convergence or even the stalling.

3) The SR1 Hessian approximation may not be positive definite along the iterations, thus resulting a direction that does not produce descent.

To prevent the method from failing due to the first drawback one simple remedy is to set $B_{k+1} = B_k$. However, this will slow down the convergence of the method. Conn, Gould and Toint (1991) and Khalfan, Byrd and Schnabel (1993) showed that the denominator of (5) vanishes rarely in practice and setting $B_{k+1} = B_k$ does not have a significant impact on the performances of the SR1 method subject to the number of the iterations or runtimes.

The second drawback is more subtle, being in close connection with the uniform linear independence of the search directions generated by the SR1 algorithm. A more precise definition of the uniform linear independence was given by Conn, Gould and Toint (1991). "A sequence $\{s_k\}$ is uniformly linearly independent if there exist $\xi > 0$, k_0 and $m \geq n$ such that, for each $k \geq k_0$, there is n distinct indices $k \leq k_1 \leq k_2 \leq \ldots \leq k_n \leq k + m$ for which the minimum singular value of the matrix $S = \left[\frac{s_{k_1}}{\|s_{k_1}\|}, \cdots, \frac{s_{k_n}}{\|s_{k_n}\|} \right]$ is at least ξ." Conn, Gould and Toint (1991) proved that the sequence of matrices generated by the SR1 formula converges to the exact Hessian, when the sequence of iterates converges to a limit point and the sequence of steps is uniformly linearly independent. Kelley and Sachs [1998] provide similar convergence results removing the first of these assumptions. Fiacco and McCormick [1968] showed that if the search directions are linearly independent and the denominator of (5) is always non-zero, then the SR1 method without line searches minimize a strongly convex quadratic function in at most $n+1$ steps. In this case B_{n+1} is exactly the Hessian of the quadratic function. Observe that this result is significant since it does not require exact line search, as is the case for the BFGS update. Generally, the above condition given by the definition of the uniform linear independency is not implemented in practice, it serves only as one of the main assumptions of a proof that the SR1 approximations to the Hessian converge to the true Hessian as the iterates converge to the solution of (1).

Subject to the uniform linear independency of the search directions Khalfan, Byrd and Schnabel [1993] showed that many problems do not satisfy this requirement, but they proved the local convergence of the SR1 method using only the positive definiteness and boundedness assumptions for the approximate Hessian. More than this Conn, Gould and Toint [1991] proved that if the minimizing function f is twice continuously differentiable and its Hessian is bounded and Lipschitz continuous and the iterates generated by the SR1 method converge to a point x^* and in addition for all k,

$$\left|(y_k - B_k s_k)^T s_k\right| \geq \xi \|y_k - B_k s_k\| \|s_k\|, \tag{9}$$

for some $\xi \in (0,1),$ and the steps s_k are uniformly linearly independent, then

$$\lim_{k\to\infty}\left\|B_k - \nabla^2 f(x^*)\right\| = 0.$$

Often condition (9) is used in implementations of the SR1 method in order to ensure that this update is well behaved. If this condition is not satisfied, then the update is skipped. Conn, Gould and Toint [1991] and Khalfan, Byrd and Schnabel [1993] provide theoretical and computational results, respectively, that if the uniform linear independence assumption is satisfied, then the approximations to the Hessian generated by the SR1 method are more accurate than those generated by BFGS, and SR1 converge faster to the true Hessian than BFGS. Therefore, if all these above drawbacks are addressed in a reliable and efficient manner, then SR1 can be used for solving (1) instead of the rank-two updates.

3 The Scaled Memory-less Quasi-Newton SR1 Update

Andrei (2021) introduced the *scaled memory-less SR1 update* by considering in (6) $H_k = I$, i.e.

$$H_{k+1} = I + t_k \frac{(s_k - y_k)(s_k - y_k)^T}{(s_k - y_k)^T y_k}, \tag{10}$$

where t_k is the scaling parameter. After some simple algebraic manipulations the *scaled memory-less SR1 search direction* $d_{k+1} = -H_{k+1}g_{k+1}$ is obtained as

$$d_{k+1} = -g_{k+1} - t_k \frac{(s_k - y_k)^T g_{k+1}}{(s_k - y_k)^T y_k}(s_k - y_k). \tag{11}$$

If $t_k = 1,$ then the *memory-less SR1 search direction* is obtained. Of course the memory-less SR1 search direction (with $t_k = 1$, i.e. not scaled) is plagued by the lack of guarantee of positive definiteness for the Hessian estimate (10). Therefore, the scaling parameter t_k is particularly introduced to ensure the positive definiteness of (10), i.e. to ensure that the search direction (11) is a descent well defined one.

The main advantage of the scaled memory-less SR1 search direction (11) is that only two scalar products $(s_k - y_k)^T g_{k+1}$ and $(s_k - y_k)^T y_k$ have to be computed. This is very advantageous for solving large-scale problems. Observe that in the scaled memory-less SR1 update the information on the Hessian approximation from the previous iteration is not accumulated to the current iteration. In a way the scaling parameter t_k in (10) is introduced to compensate this loss of information and as we said to ensures that the search direction (11) is a descent one.

To determine a value for the scaling parameter t_k two procedures have been developed in (Andrei, 2021). The first one is based on the sufficient descent condition the second considers the conjugacy condition from the conjugate gradient algorithms.

1) Firstly, by imposing the sufficient descent condition

$$g_{k+1}^T d_{k+1} = -c\|g_{k+1}\|^2, \tag{12}$$

where d_{k+1} is given by (11) the following value for the scaling parameter is obtained

$$t_k = \frac{(s_k - y_k)^T y_k}{[(s_k - y_k)^T g_{k+1}]^2}(c-1)\|g_{k+1}\|^2. \tag{13}$$

Observe that in (12) the classical sufficient descent condition (11) is modified with equality. In (13) c is selected close to 1, but not too close. In our numerical experiments we selected $c = 7/8$. Therefore, introducing the value for t_k given by (13) in (11) the following scaled memory-less SR1 search direction is obtained

$$d_{k+1} = -g_{k+1} - \frac{(c-1)\|g_{k+1}\|^2}{(s_k - y_k)^T g_{k+1}}(s_k - y_k). \tag{14}$$

Observe that the iterations are affected if the denominator $(s_k - y_k)^T g_{k+1}$ of the update term becomes too small. In this case the update term in (14) may start to dominate the negative gradient and therefore the influence of the negative gradient in the search direction is lost. In order to accommodate these situations, the rule we apply is that the updates are skipped whenever the denominator $(s_k - y_k)^T g_{k+1}$ is too small in the sense

$$\left|(s_k - y_k)^T g_{k+1}\right| < \eta\|s_k - y_k\|\|g_{k+1}\|, \tag{15}$$

that is, if (15) is satisfied, then $d_{k+1} = -g_{k+1}$. A typical value for η is 10^{-8}.
2) Secondly, from the conjugacy condition

$$d_{k+1}^T y_k = -h g_{k+1}^T s_k, \tag{16}$$

where $h \geq 0$ is a scalar and d_{k+1} is given by (11) the following value for the scaling parameter t_k is obtained

$$t_k = \frac{(hs_k - y_k)^T g_{k+1}}{(s_k - y_k)^T g_{k+1}}. \tag{17}$$

Now, introducing the value for t_k given by (17) in (11) the following scaled memory-less SR1 search direction is obtained

$$d_{k+1} = -g_{k+1} - \frac{(hs_k - y_k)^T g_{k+1}}{(s_k - y_k)^T y_k}(s_k - y_k). \tag{18}$$

In our numerical experiments we selected $h = 0.5$. Again observe that the iterations are affected if the denominator $(s_k - y_k)^T y_k$ of the update term is too small. In these cases if

$$\left|(s_k - y_k)^T y_k\right| < \eta \|s_k - y_k\| \|y_k\|, \tag{19}$$

with $\eta = 10^{-8}$, then $d_{k+1} = -g_{k+1}$. In Andrei (2021) it is proved that the scaled memory-less SR1 search direction (18) computed from the conjugacy conditions is a descent direction. Besides, numerical test for solving 800 unconstrained optimization problems with different structures and complexities with the number of variables in the range [100, 1000] showed that the scaled memory-less SR1 method based on sufficient descent or on conjugacy conditions are more efficient and more robust versus BFGS from CONMIN [Shano, 1983; Shanno and Phua, 1976], one of the best implementation of BFGS.

4 SR1 with Cubic Regularization

The key feature of the cubic regularization is the computation of the step from the current iteration to the next one by minimizing a cubic over-estimator of the minimizing function. As we know, the step direction d computed by the Newton method minimizes

$$f_N(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d.$$

If the Hessian of the minimizing function f is Lipschitz continuous with constant L, then the Taylor expansion of f around the current point x_k gives

$$\begin{aligned} f(x_k + d) &= f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d + \int_0^1 (1-\tau) d^T (\nabla^2 f(x_k + \tau d) - \nabla^2 f(x_k)) d \mathrm{d}\tau \\ &\le f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d + \frac{L}{6} \|d\|^3 \triangleq f_L(x_k + d), \end{aligned} \tag{20}$$

for all $d \in \mathbb{R}^n$. Therefore, using this estimation we can get a method for solving (1). However, the main deficiency of this approach is that for general nonlinear optimization problems there is no algorithm for explicit determination of L. Instead, better is to consider a positive approximation M to the Lipschitz constant, known as regularization parameter, and define

$$f_M(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d + \frac{M}{6} \|d\|^3. \tag{21}$$

Therefore, the cubic step direction is obtained as solution of the following subproblem

$$d = \arg\min_s f_M(x_k + s), \tag{22}$$

which is exactly the Adaptive Regularization algorithm using Cubics (ARC) suggested by Cartis, Gould and Toint (2011a, b). Two problems are present with this approach based on minimizing (21). The first one is how to choice the regularization parameter M. The second one is how to solve the nonlinear optimization problem (22). A simple approach to choice M is to start with a very small positive value and then to increase it to be sufficiently large. For the second problem, Cartis, Gould and Toint (2011a, b) propose solving (22) only approximately. They prove that the resulting algorithm retain the convergence properties described by Nesterov and Polyak (2006).

4.1 Modified SR1 Update Formula

For the very beginning observe that the secant equation (4) is associated with a quadratic model of the minimizing function in a neighborhood of x_k, i.e. it is associated to f_N. Besides, this equation is satisfied exactly only when B_{k+1} is equal to the true Hessian and the minimizing function is a quadratic. Benson and Shanno (2018) introduced a modified secant equation using for this the function f_M instead of f_N. In this case the secant equation (4) is replaced with $B_{k+1}s_k^M = y_k^M$, where from (21) we have

$$s_k^M = x_{k+1} - x_k,$$
$$y_k^M = \nabla f_M(x_{k+1}) - \nabla f_M(x_k) = y_k + \frac{M}{2}\|s_k\| s_k.$$

Therefore, substituting the above elements in (4) the *new secant equation* is

$$y_k = \left(B_k - \frac{M}{2}\|s_k\| I \right) s_k. \tag{23}$$

Now, to get a new rank-one formula for updating the approximation B_k is to find the scalar γ and the vector $u \in \mathbb{R}^n$ such that $B_{k+1} = B_k + \gamma u u^T$. Substituting this expression into the secant equation we get

$$y_k = \left(B_k + \gamma u u^T - \frac{M}{2}\|s_k\| I \right) s_k$$

But, $u u^T s_k = (s_k^T u)u = (u^T s_k)u.$ With this, we get

$$y_k - \left(B_k - \frac{M}{2}\|s_k\| I \right) s_k = \gamma (u^T s_k) u.$$

Since $\gamma(u^T s_k)$ is a scalar, in order to satisfy this equation we can select $u = y_k - \left(B_k - \frac{M}{2}\|s_k\| I \right) s_k$ and $\gamma = (u^T s_k)^{-1}$. Therefore, the following *modified SR1 updating formula* for approximation to the Hessian is obtained

$$B_{k+1} = B_k + \frac{[y_k - (B_k - \frac{M}{2}\|s_k\| I)s_k][y_k - (B_k - \frac{M}{2}\|s_k\| I)s_k]^T}{[y_k - (B_k - \frac{M}{2}\|s_k\| I)s_k]^T s_k}. \quad (24)$$

Using the Sherman-Morrison formula to (24), the following modified SR1 updating formula for approximation to the inverse Hessian is obtained

$$H_{k+1} = H_k + \frac{[s_k - H_k(y_k + \frac{M}{2}\|s_k\| s_k)][s_k - H_k(y_k + \frac{M}{2}\|s_k\| s_k)]^T}{[s_k - H_k(y_k + \frac{M}{2}\|s_k\| s_k)]^T (y_k + \frac{M}{2}\|s_k\| s_k)}. \quad (25)$$

Observe that if $M = 0,$ then the SR1 updating formulae (5) and respectively (6) are obtained.

4.2 Modified Memory-less SR1 Update Formula

This update is obtained by considering $B_k = I$ in (24) or $H_k = I$ in (25). Therefore considering $H_k = I$ in (25) we get

$$H_{k+1} = I + \frac{[s_k - (y_k + \frac{M}{2}\|s_k\| s_k)][s_k - (y_k + \frac{M}{2}\|s_k\| s_k)]^T}{[s_k - (y_k + \frac{M}{2}\|s_k\| s_k)]^T (y_k + \frac{M}{2}\|s_k\| s_k)} \quad (26)$$

which is the modified memory-less SR1 updating formula for approximation to the inverse Hessian. The *modified memory-less SR1 search direction* is computed as $d_{k+1} = -H_{k+1}g_{k+1},$ where H_{k+1} is given by (26), i.e.

$$d_{k+1} = -g_{k+1} - \frac{[s_k - (y_k + \frac{M}{2}\|s_k\| s_k)]^T g_{k+1}}{[s_k - (y_k + \frac{M}{2}\|s_k\| s_k)]^T (y_k + \frac{M}{2}\|s_k\| s_k)}[s_k - (y_k + \frac{M}{2}\|s_k\| s_k)]. \quad (27)$$

Denote $v_k = y_k + \frac{M}{2}\|s_k\| s_k.$ With this, the search direction (27) corresponding to the modified memory-less SR1 updated formula can be written as

$$d_{k+1} = -g_{k+1} - \frac{(s_k - v_k)^T g_{k+1}}{(s_k - v_k)^T v_k}(s_k - v_k). \quad (28)$$

The numerator of H_{k+1} from (26) is a symmetric, rank-one, positive definite matrix for any M. Therefore, in order to have H_{k+1} a positive definite matrix, we have to choose M in such a way that the denominator of H_{k+1} is strictly positive, i.e.

$$[(s_k - y_k) - \frac{M}{2}\|s_k\| s_k]^T (y_k + \frac{M}{2}\|s_k\| s_k) > 0. \tag{29}$$

After some simple algebraic manipulations (29) can be written as $aM^2 + bM + c > 0$, where

$$a = -\frac{\|s_k\|^4}{4}, \quad b = \frac{\|s_k\|^3}{2} - \|s_k\|(y_k^T s_k), \quad c = y_k^T s_k - \|y_k\|^2. \tag{30}$$

Define $\varphi(M) = aM^2 + bM + c$. Observe that the denominator of (26) is exactly $\varphi(M)$, which is a quadratic concave function with positive values between its roots, if they exist. Observe that $a < 0$. Now, since c is the denominator of the memory-less SR1 update it follows that $c < 0$. Therefore, $4ac > 0$. On the other hand, b can be positive or negative. If they exist, the roots of the equation $\varphi(M) = 0$ are

$$M_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Therefore, taking into consideration the quadratic function $\varphi(M)$, in order to have a well defined modified memory-less SR1 updating formula (26), the positive regularization parameter M may be selected as follows.

Case 1	If $b^2 - 4ac < 0$, then the quadratic function $\varphi(M)$ does not have real roots. In this case the highest value attained by $\varphi(M)$ is negative, i.e. $-b/2a < 0$. Therefore, in this case there is no value for M for which the modified memory-less updated matrix H_{k+1} is positive definite. Set $M = 0$.
Case 2	If $b^2 - 4ac \geq 0$ and $b < 0$, then both roots of the quadratic $\varphi(M)$ are negative, or one is negative and the other one is positive. If both roots of $\varphi(M)$ are negative, then set $M = 0$, i.e. there is no positive value of M for which H_{k+1} is positive definite. If one root, let say $M_1 < 0$, and the other one, let say $M_2 > 0$, then set M to any value in interval $(0, M_2)$.
Case 3	If $b^2 - 4ac \geq 0$ and $b > 0$, then both roots of the quadratic $\varphi(M)$ are positive, or one is negative and the other one is positive. If both roots of $\varphi(M)$ are positive, then set $M = -b/2a$ to get the highest positive value of the quadratic which corresponds to the largest value of the denominator of (26). If one root, let say $M_1 < 0$, and the other one, let say $M_2 > 0$, then set M to any value in interval $(0, M_2)$.

We emphasize that if $M = 0$, then we can not guarantee a positive definite update matrix H_{k+1}, given by (26), with this modified memory-less SR1 formula. In other words, H_{k+1} given by (26) with $M = 0$ reduces to

$$H_{k+1} = I + \frac{(s_k - y_k)(s_k - y_k)^T}{(s_k - y_k)^T y_k},$$

which is plagued by the lack of guarantee of positive definiteness. In fact, the scaling parameter t_k was introduced in the scaling memoryless SR1 updating formula (10) as a technical artifice to get a positive definite scaled memory-less updating matrix, as was discussed in Section 3 above. Therefore, in these cases, when $M = 0,$ we suggest using some strategies as follows. One is to consider as search direction the negative gradient instead of (28), i.e. $d_{k+1} = -g_{k+1}$. This is not the best selection, as it is equivalent to reverting to the steepest descent, which as we know converges very slow. However, selection of negative gradient is motivated by the fact that even Newton methods for nonconvex nonlinear optimization use a steepest descent step when negative curvature is encountered. The other one, we adopt in our numerical experiments, is to consider the search directions corresponding to the scaled memory-less SR1 update (10) in which the parameter t_k is computed by imposing the sufficient descent condition (12), or from the conjugacy condition (16). Anyway, when $M = 0$ the cubic regularization is abandoned and we revert to the scaled memory-less SR1 update formula. In order to implement this strategy the algorithm includes two logical parameters: *suff* and *conj*. If $M = 0$ and *suff* = *.true.*, then the scaled memory-less SR1 search direction based on sufficient descent condition given by (14) and (15) is used. On the other hand if $M = 0$ and *conj* = *.true.*, then the scaled memory-less SR1 search direction based on conjugacy condition given by (18) and (19) is used.

When the above cases determine $M > 0$, then several variants for its computation have been implemented and tested. All of them refer to the selection of a value of M in the interval defined by zero and by the positive root of the quadratic $\varphi(M),$ when it exists.

In Case 1, if $b^2 - 4ac < 0,$ there is no positive value for the regularization parameter M for which the modified memory-less SR1 approximation to the inverse Hessian is positive definite. However, it is quite possible that $|b^2 - 4ac| \le \varepsilon_r,$ where $\varepsilon_r > 0$ is a sufficiently small parameter and $b > 0.$ In this case, since $a < 0$ we can set $M = -b/2a.$ In our numerical experiments we selected $\varepsilon_r = 10^{-8}.$ However, this selection of M does not proved to be benefic subject to the performances of the algorithm.

In Cases 2 and 3, as we noticed, there is more room for selecting a positive value for M in the interval $(0, M_2).$ Our selection is more conservative in sense that we selected M as the middle of the interval.

It is worth mentioning that Case 2 arises more often than Case 3, but this is not a definitive remark.

Proposition 1 *The modified memory-less SR1 search direction with cubic regularization* (28) *is a descent direction.*

Proof Since the parameter M is selected in such a way that $(s_k - v_k)^T v_k > 0,$ from (28) we have that

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 - \frac{[g_{k+1}^T(s_k - v_k)]^2}{(s_k - v_k)^T v_k} \leq 0,$$

i.e. the search direction (28) is a descent one. ♦

With these developments the following accelerated modified memory-less SR1 with cubic regularization algorithm may be presented.

Algorithm AMMSR1

1.	Initialization. Consider an initial point x_0. Set $k = 0$. Select some values for the Wolfe line search conditions σ and ρ with $0 < \rho < \sigma < 1$. Compute $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$. Select the sufficiently small parameters $\varepsilon > 0$ used in the criterion for stopping the iterations and $\varepsilon_A > 0$ used in the acceleration scheme. Select some positive values for parameters c and h. Set the logical variables *conj* or *suff* on .true. or on .false.
2.	Test a criterion for stopping the iterations: if $\|g_k\|_\infty \leq \varepsilon$ then stop the iterations, otherwise go to step 3
3.	Compute the stepsize α_k using the standard Wolfe line search conditions
4.	Update the variables $x_{k+1} = x_k + \alpha_k d_k$ and compute f_{k+1} and g_{k+1}. Compute $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$
5.	Acceleration scheme: a) Compute: $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_k = g_k - g_z$ b) Compute: $\bar{a}_k = \alpha_k g_k^T d_k$, and $\bar{b}_k = -\alpha_k y_k^T d_k$ c) If $\|\bar{b}_k\| \geq \varepsilon_A$, then compute $\xi_k = -\bar{a}_k / \bar{b}_k$ and update the variables as $x_{k+1} = x_k + \xi_k \alpha_k d_k$. Compute f_{k+1} and g_{k+1}. Compute $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$
6.	Compute the regularization parameter M
7.	If $M = 0$, and if *suff* = .true., then compute the scaled memory-less SR1 search direction based on the sufficient descent condition, as in (14) with (15) If $M = 0$, and if *conj* = .true., then compute the scaled memory-less SR1 search direction based on the conjugacy condition, as in (18) with (19)
8.	If $M > 0$, compute $v_k = y_k + \frac{M}{2}\|s_k\| s_k$ and then compute the search direction $d_{k+1} = -g_{k+1} - \frac{(s_k - v_k)^T g_{k+1}}{(s_k - v_k)^T v_k}(s_k - v_k)$
9.	Restart iterations. If $g_{k+1}^T d_{k+1} > -10^{-3}\|g_{k+1}\|\|d_{k+1}\|$, then set $d_{k+1} = -g_{k+1}$
10.	Consider $k = k + 1$ and go to step 2 ♦

Observe that the algorithm is equipped with an acceleration scheme (see step 5) introduced by Andrei (2006, 2009). This scheme modifies the stepsize determined by the Wolfe line search conditions (3) in such a way as to improve the reduction of the minimizing function values along the iterations. It is proved that this acceleration scheme is linear convergent, but the reduction in the function value is significantly improved.

If f is bounded along the direction d_k, then there exists a stepsize α_k satisfying the Wolfe line search conditions (3). The first trial of the stepsize crucially affects the practical behavior of the algorithm. At every iteration $k \geq 1$, the starting guess for the step α_k in the line search is computed as $\alpha_{k-1}\|d_{k-1}\|/\|d_k\|$.

In our algorithm we introduced a restarting condition. If this restarting condition is satisfied, then the algorithm is restarted with the negative gradient. Some other restarting procedures may be implemented, but we are interested in seeing the performances of AMMSR1 implementing this restart condition.

5 Convergence of the Algorithm

In this section the global convergence of the algorithm is established under the following assumptions:

(A1) The level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded, i.e. there exists a constant $B > 0$ such that for any $x \in \Omega$, $\|x\| \leq B$.

(A2) The function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its gradient is Lipschitz continuous in a neighborhood N of Ω, i.e. there exists a constant $L > 0$ such that $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, for any $x, y \in N$.

It is easy to see that under these assumptions, there exists a constant $\Gamma > 0$ such that $\|\nabla f(x)\| \leq \Gamma$, for any $x \in \Omega$.

Although the search directions d_{k+1} generated by the algorithm AMMSR1 are always descent directions, in order to get the convergence of the algorithm we need to derive a lower bound for the stepsize α_k.

Proposition 2 *Suppose that d_k is a descent direction and ∇f satisfies the Lipschitz condition*

$$\|\nabla f(x) - \nabla f(x_k)\| \leq L\|x - x_k\|$$

for all x on the line segment connecting x_k and x_{k+1}, where L is a constant. If the line search satisfies the standard Wolfe conditions (3)*, then*

$$\alpha_k \geq \frac{1-\sigma}{L}\frac{\left|g_k^T d_k\right|}{\|d_k\|^2}. \tag{31}$$

Proof Subtracting $g_k^T d_k$ from both sides of (3.b) and using the Lipschitz condition, it follows that

$$(\sigma - 1)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq \alpha_k L\|d_k\|^2.$$

But d_k is a descent direction and $\sigma < 1$, therefore (31) follows from the above inequality. ♦

Proposition 3 *Suppose that* $(s_k - v_k)(s_k - v_k)^T$, where $v_k = y_k + \frac{M}{2}\|s_k\| s_k$, *is bounded above in norm. Then the modified memory-less SR1 update matrix* H_{k+1} *with cubic regularization given by* (26) *is bounded above in norm.*

Proof The proof follows the three cases presented above. In the first case H_{k+1} is updated with $M = 0$. Therefore, since H_k and $(s_k - v_k)(s_k - v_k)^T$ are bounded above, it follows that H_{k+1} is bounded above. In Cases 2 and 3, H_{k+1} is updated using (26) with the regularization parameter M computed in such a way that the denominator of the update matrix is strictly greater than zero. In fact the value of M is selected in the open interval given by zero and the positive root of the quadratic function $\varphi(M)$. Therefore, the denominator of (26) is sufficiently larger than zero. Now, since the matrix $(s_k - v_k)(s_k - v_k)^T$ is bounded above in norm and $\|s_k\| = \|x_{k+1} - x_k\| \le \|x_{k+1}\| + \|x_k\| \le 2B$, (see assumption (A1)), it follows that the numerator of H_{k+1} given by (26) is bounded above as well. Therefore, the modified memory-less SR1 update matrix H_{k+1} is bounded above in norm. ♦

Proposition 4 *For any iteration* k, *the modified memory-less SR1 search direction* (27), *is a descent direction, that is* $g_{k+1}^T d_{k+1} < 0$. *Besides,* $\|d_{k+1}\| \le D$, *for some* $D > 0$.

Proof Observe that the numerator of the update (26) is a rank-one positive semidefinite matrix and the regularization positive parameter M is chosen so that the denominator of (26) is strictly positive. Therefore the update matrix H_{k+1} given by (26) is positive semidefinite. Without loss of generality, H_k is positive definite, hence, H_{k+1} given by (26) is positive definite. Thus, $g_{k+1}^T d_{k+1} = -g_{k+1}^T H_{k+1} g_{k+1} < 0$.
The boundedness of d_{k+1} may be established as follows. If the restart condition is satisfied, then $d_{k+1} = -g_{k+1}$. Therefore, $\|d_{k+1}\| = \|g_{k+1}\| < D$, where $D = \Gamma$ by assumption (A2). On the other hand, for non-restart iterations the search direction is computed as $d_{k+1} = -H_{k+1} g_{k+1}$, where H_{k+1} is given by (26). In this case, by Proposition 3 the sequence of the Hessian matrices given by (26) remains bounded above in norm. Therefore, $\|d_{k+1}\| = \|-H_{k+1} g_{k+1}\| \le \|H_{k+1}\| \|g_{k+1}\|$, which is bounded above by some constant $D > 0$. ♦

Theorem 1 *Suppose that the assumptions (A1) and (A2) are satisfied. If* f *is Lipschitz continuous, then* $\lim_{k \to \infty} \|g_k\| = 0$.

Proof The assumptions and the above Propositions 2, 3 and 4 show that the algorithm AMMSR1 is globally convergent to a point in which the first-order optimality conditions are satisfied. ♦

6 Numerical Examples

In this section we report some numerical results obtained with an implementation of the AMMSR1 algorithm. The code is written in Fortran and compiled with f77 (default compiler settings) on a Workstation Intel Pentium 4 with 1.8 GHz. We selected a number of 80 large-scale unconstrained optimization test functions in generalized or extended form we presented in [Andrei, 2020]. The vast majority of these problems are taken from CUTE collection [Bongartz, Conn, Gould and Toint, 1995]. For each test function we have taken ten numerical experiments with the number of variables increasing as $n = 1000, 2000, ..., 10000$. Therefore, a number of 800 unconstrained optimization problems have been considered. The algorithms used in all numerical test implement the Wolfe line search conditions (3) with $\rho = 0.0001$, $\sigma = 0.8$ and the same stopping criterion $\|g_k\|_\infty \leq 10^{-6}$, where $\|.\|_\infty$ is the maximum absolute component of a vector. In all algorithms, we considered in our numerical studies, the maximum number of iterations is limited to 10000, while the maximum number of function and its gradient evaluations is limited to 10000. In all codes the stepsize α_k is computed as in subroutine CONMIN (Shanno, 1983), which generates safeguarded stepsizes satisfying the Wolfe line search condition (3) by building a cubic model of the minimizing function and trying to minimize the cubic model by stopping at any iteration that gives a sufficient reduction of the minimizing function subject to the Armijo condition (3.a).

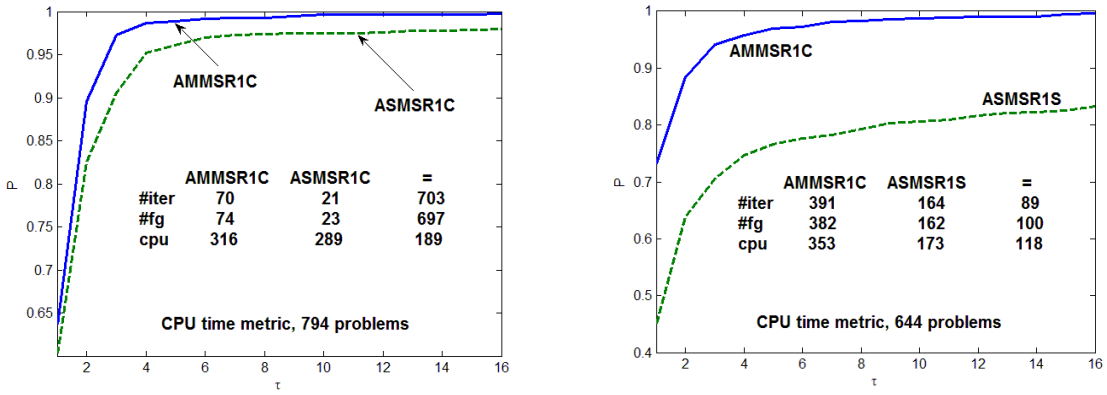
The comparisons of algorithms are given in the following context. Let f_i^{ALG1} and f_i^{ALG2} be the optimal value found by ALG1 and ALG2, for problem $i = 1, \ldots, 800$, respectively. We say that, in the particular problem i, the performance of ALG1 was better than the performance of ALG2 if:

$$\left| f_i^{ALG1} - f_i^{ALG2} \right| < 10^{-3} \tag{32}$$

and the number of iterations (#iter), or the number of function-gradient evaluations (#fg), or the CPU time of ALG1 was less than the number of iterations, or the number of function-gradient evaluations, or the CPU time corresponding to ALG2, respectively.

The general algorithm AMMSR1 may be particularized as: AMMSR1C – the accelerated modified memory-less SR1 with cubic regularization using the conjugacy condition, i.e. with *conj* = .true. and AMMSR1S - the accelerated modified memory-less SR1 with cubic regularization using the sufficient descent condition, i.e. with *suff* = .true. These algorithms are compared versus ASMSR1C – the accelerated scaled memory-less SR1 with conjugacy condition in which the search direction is computed as in (18) with (19) and versus ASMSR1S – the accelerated scaled memory-less SR1 with sufficient descent condition in which the search direction is computed as in (14) with (15).

In the first set of numerical experiments AMMSR1C is compared versus ASMSR1C and versus ASMSR1S for solving 800 unconstrained optimization problems with the number of variables in the range [1000, 10000]. Figure 1 presents the Dolan and Moré (2002) performance profiles of these algorithms subject to the CPU computing time.



a) **b)**

Fig. 1. Performance profiles of AMMSR1C versus ASMSR1C and versus ASMSR1S. $n \in [1000,10000]$

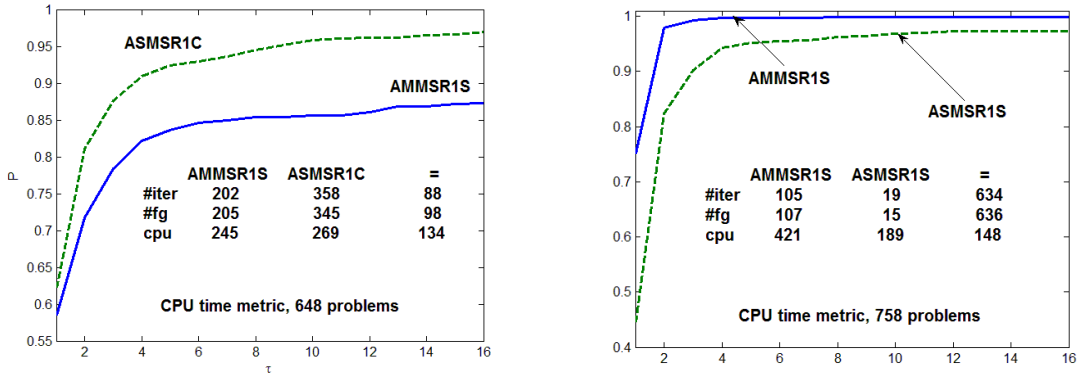
In a performance profile plot, the top curve corresponds to the method that solved the most problems in a time that was within a given factor of the best time. The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right side of the plot gives the percentage of the test problems that were successfully solved by these algorithms, respectively. Mainly the left side of the plot is a measure of the efficiency of an algorithm, while the right side is a measure of the robustness of an algorithm.

Comparing AMMSR1C versus ASMSR1C, (see Figure 1.a) subject to the number of iterations, we see that AMMSR1C was better in 70 problems (i.e. it achieved the minimum number of iterations in 70 problems). ASMSR1C was better in 21 problems and they achieved the same number of iterations in solving 703 problems, etc. Out of 800 problems considered in this numerical experiment, only for 794 problems does the criterion (32) hold. Observe that the differences are significant. Subject to the CPU time metric, AMMSR1C was faster in 316 problems, while ASMSR1C was faster only in 289, i.e. AMMSR1C is more efficient than ASMSR1C. The accelerated modified memory-less SR1 with cubic regularization and conjugacy condition is more efficient and more robust than the accelerated scaled memory-less SR1 method based on conjugacy condition. Notice that both AMMSR1C and ASMSR1C use the same implementation of the standard Wolfe line search (3) based on cubic interpolation (Shanno, 1983), as well as the same optimization conditions.

Comparing AMMSR1C versus ASMSR1S (see Figure 1.b) we observe that the accelerated modified memory-less SR1 method with cubic regularization and conjugacy condition is more efficient and more robust than the accelerated scaled memory-less SR1 method based on sufficient descent condition. The differences between these algorithms are significant. For example, subject to the CPU time metric, AMMSR1C was fastest for solving 353 problems, while ASMSR1S was fastest only in 173 problems, etc. Anyway,

cubic regularization ensures sufficient descent at each iteration by controlling in an efficient manner the approximation to the Lipschitz constant M rather than to impose a stepsize α_k as in the Newton method.

In the second set of numerical experiments AMMSR1S is compared versus ASMSR1C and versus ASMSR1S. Figure 2 shows the performance profiles of these algorithms subject to the CPU computing time.



a) **b)**

Fig. 2. Performance profiles of AMMSR1S versus ASMSR1C and versus ASMSR1S $n \in [1000,10000]$

From Figure 2.a we see that the accelerated scaled memory-less SR1 method with conjugacy condition, ASMSR1C, is more efficient and more robust than the accelerated modified memory-less SR1 with cubic regularization and using sufficient descent condition, AMMSR1S. The scaled memory-less SR1 search direction computed from the conjugacy conditions is a descent direction (Andrei, 2021). In the economy of an optimization algorithm the conjugacy condition is an important ingredient. The main characteristic of conjugate directions is that to minimize a convex quadratic function in a subspace spanned by a set of mutually conjugate directions is equivalent to minimize the function along each conjugate direction in turn. However, the performances of algorithms satisfying the conjugacy condition are strongly dependent on the accuracy of the line search. If the line search procedure for stepsize computation is highly accurate, then the corresponding optimization algorithm is extremely fast.
In Figure 2.b we have a numerical confirmation that the accelerated modified memory-less SR1 with cubic regularization using sufficient descent condition, AMMSR1S, is more efficient and more robust than the accelerated scaled memory-less SR1 method with sufficient descent, ASMSR1S. Both algorithm AMMSR1S and ASMSR1S are based on the sufficient descent condition. However, AMMSR1S implements cubic regularization through the modified secant equation.

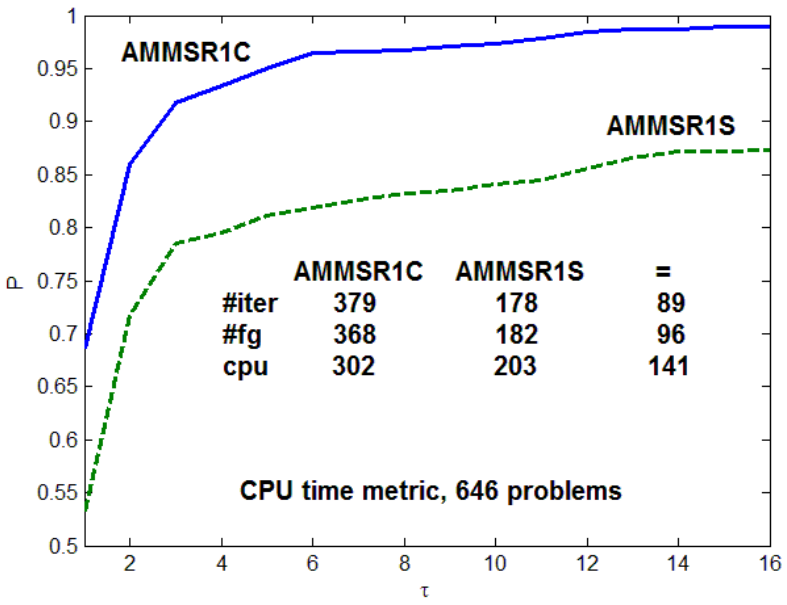


Fig. 3. Performance profiles of AMMSR1C versus AMMSR1S.
$n \in [1000,10000]$

It is worth comparing AMMSR1C versus AMMSR1S. Figure 3 presents the performance profiles of these algorithms. Observe that the accelerated modified memory-less SR1 method based on cubic regularization with conjugacy condition is way more efficient and more robust than the accelerated modified memory-less SR1 methods based on cubic regularization with sufficient descent condition. Both these algorithms implement the same line search procedure and both of them generate descent search directions (Andrei, 2021). However, this is not sufficient to get highly performer algorithms. In this case, it is the conjugacy condition that makes the difference.

We emphasize here that SR1 method (5) is obtained by updating an approximation to the Hessian of the minimizing function with a rank-one matrix computed by using the classical secant equation. On the other hand, the modified SR1 method (24) is obtained by updating an approximation to the Hessian with a rank-one matrix computed this time by using the modified secant equation (23). This modified secant equation which includes an estimation M of the Lipschitz constant L of the Hessian determines a better updating term in (24), leading us to better approximations to the true Hessian. Like in (Benson and Shanno, 2018) in this paper the estimation M of the Lipschitz constant L of the Hessian is obtained by cubic regularization. It is worth saying that the memory-less technique used here in an intensive manner is one of the simplest methods to adapt the known optimization methods for solving large-scale problems. In the frame of the quasi-Newton methods, by memory-less technique we understand that, at every iteration, the current approximation to the Hessian is obtained by updating the identity matrix. This technique was for the first time introduced by Shanno (1978, 1978) who proved that the conjugate gradient methods are precisely the BFGS quasi-Newton method, where the approximation to the inverse Hessian is restarted as the identity matrix at every iteration.

From the above numerical experiments we see that the cubic regularization of the quadratic model of the minimizing function is more robust than the scaling of the memory-less SR1 updating term.

In the third set of numerical experiments let us compare accelerated modified memory-less SR1 method based on cubic regularization with conjugacy condition, AMMSR1C, and the accelerated modified memoryless SR1 method based on cubic regularization with sufficient descent, AMMSR1S, versus BFGS implemented in CONMIN (Shanno and Phua, 1976) using the same set of 800 unconstrained optimization problems with the number of variables in the range [100-1000]. Figure 4 shows the performance profiles of these algorithms.

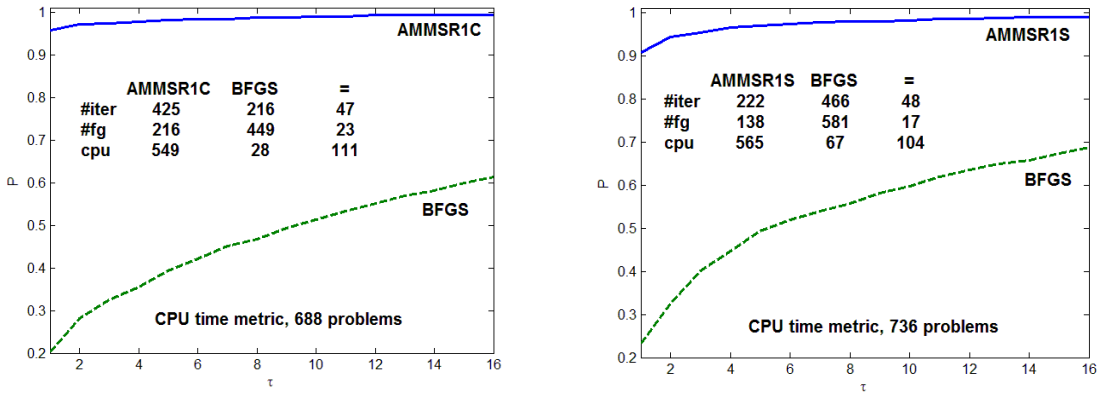


Fig. 4. Performance profiles of AMMSR1C versus BFGS and of AMMSR1S versus BFGS. $n \in [100,1000]$

From Figure 4 we see that subject to the CPU computing time both AMMSR1C and AMMSR1S are way more efficient and more robust than the BFGS in implementation of CONMIN. For example, subject to CPU time metric, AMMSR1C was better in solving 549 problems, while BFGS was better in solving only 28 problems, etc. There is a great difference between these algorithms. The BFGS in CONMIN is a variable metric method with initial scaling which approximately needs $n^2/2+11n/2$ double precision words of working storage. In comparison AMMSR1 requires approximately $7n$ double precision words of working storage. BFGS requires more memory and involves a greater computational effort.

In the fourth set of numerical experiments let us present comparisons between AMMSR1C versus ASMSR1C and between AMMSR1S versus ASMSR1S algorithms for solving some applications from the MINPACK-2 test problem collection [Averick, Carter, Moré, & Xue, 1992]. The minimizing function of all these applications is quadratic. In Table 1, we present these applications, as well as the values of their parameters. The infinite-dimensional version of these problems is transformed into a finite element approximation by triangulation. Thus a finite-dimensional minimization problem is obtained whose variables are the values of the piecewise linear function at the

vertices of the triangulation. The discretization steps are $nx = 200$ and $ny = 200$, thus obtaining minimization problems with $nx \times ny = 40{,}000$ variables.

Table 1
Applications from the MINPACK-2 collection

A1	Elastic–plastic torsion [Glowinski, 1984, pp. 41–55], c = 5
A2	Pressure distribution in a journal bearing [Cimatti, 1977], b = 10, ε = 0.1
A3	Optimal design with composite materials [Goodman, Kohn, & Reyna, 1986], λ = 0.008
A4	Steady-state combustion [Aris, 1975, pp. 292–299], [Bebernes, & Eberly, 1989], λ = 5
A5	Minimal surfaces with Enneper conditions [Nitsche, 1989, pp. 80–85]

The performances of the accelerated modified memory-less SR1 method based on cubic regularization with conjugacy condition, AMMSR1C, versus the accelerated scaled memory-less SR1 method based on conjugacy condition, ASMSR1C, are given in Table 2, where #iter is the number of iterations, #fg is the number of function and its gradient evaluations and cpu is the CPU time computing.

Table 2
Performances of AMMSR1C versus ASMSR1C (40,000 variables, CPU seconds)

	AMMSR1C			**ASMSR1C**		
	#iter	#fg	cpu	#iter	#fg	cpu
A1	13138	26297	277.10	13138	26297	273.64
A2	66922	133876	1476.36	66922	133876	1608.61
A3	87482	175001	3498.05	89559	179187	3430.25
A4	49631	99287	3964.23	49631	99287	3815.94
A5	11374	22784	285.36	11374	22784	300.46
TOTAL	**228547**	**457245**	**9501.10**	**230624**	**461431**	**9428.90**

The performances of the accelerated modified memory-less SR1 method based on cubic regularization with sufficient descent, AMMSR1S, versus the accelerated scaled memory-less SR1 method based on sufficient descent, ASMSR1S, are given in Table 3.

Table 3
Performances of AMMSR1S versus ASMSR1S (40,000 variables, CPU seconds)

	AMMSR1S			**ASMSR1S**		
	#iter	#fg	cpu	#iter	#fg	cpu
A1	514	1314	12.28	1197	3061	32.19
A2	12306	26343	288.34	9691	20279	228.76
A3	3587	8809	136.30	6665	16587	295.33
A4	941	2436	103.98	1715	4394	172.60
A5	627	1592	20.01	718	1831	20.83
TOTAL	**17975**	**40494**	**560.91**	**19986**	**46152**	**749.71**

Table 4 presents the characteristics of the optimization process for AMMSR1C and AMMSR1S, respectively. In these tables #conj is the number of iterations in which the

search direction is computed as the scaled memory-less SR1 search direction given by (18) with (19) using the conjugacy condition. #suff is the number of iterations in which the search direction is computed as the scaled memory-less SR1 search direction given by (14) with (15) using the sufficient descent. On the other hand, #grad represents the number of iterations in which the search direction is the negative gradient.

Table 4
Characteristics of AMMSR1C versus AMMSR1S (40,000 variables, CPU seconds)

	AMMSR1C			**AMMSR1S**		
	#iter	#conj	#grad	#iter	#suff	#grad
A1	13138	13137	1	514	498	16
A2	66922	66919	3	12306	3648	8658
A3	87482	87478	4	3587	3561	26
A4	49631	49630	1	941	909	32
A5	11374	11372	2	627	613	14
TOTAL	**228547**	**228536**	**11**	**17975**	**9229**	**8746**

From the above tables we see that both the scaled memory-less SR1 and the modified memory-less SR1 with cubic regularization algorithms are able to solve large-scale unconstrained optimization problems. Both AMMSR1C and ASMSR1C have similar performances. However, subject to the number of iterations, to the number of function and its gradient evaluations and to the CPU computing time, AMMSR1S is more efficient than ASMSR1S. It seems that cubic regularization with sufficient descent condition is an important ingredient for improving the performances of SR1 method.

Observe that, subject to the CPU time metric, the scaled memory-less SR1 method with sufficient descent and the modified memory-less SR1 method also with sufficient descent are faster than the scaled memory-less SR1 method with conjugacy condition and the modified memory-less SR1 method also with conjugacy condition. In other words, AMMSR1S is more efficient than AMMSR1C. Obviously, sufficient descent condition is more important than conjugacy condition.

From Table 4 we see that in case of AMMSR1C the search direction given by the negative gradient very rare is used. Out of 228547 iterations for solving all 5 applications, only in 11 (i.e. 0.0048%) iterations the negative gradient was used. In contrast, in case of AMMSR1S we see that out of 17975 iterations for solving all 5 applications, only in 8746 (i.e. 48.66%) iterations the negative gradient was used. Of course, excepting the application A2, in case of AMMSR1S, also the negative gradient is used in a very small number of iterations. It is worth saying that for solving large-scale optimization problems the best performances are obtained by those algorithms for which for a small fraction of iterations the negative gradient is used. AMMSR1S satisfies this tacit custom.

7 Conclusions

The paper presents new variants for SR1 update based on the memory-less technique and on a modified secant equation obtained from the cubic overestimation of the minimizing function. The idea of using these techniques was to get a positive definite update of the SR1 method able for solving large-scale minimization problems. When is not possible to

get a strictly positive value for the cubic regularization parameter, then the modified memory-less SR1 with cubic regularization reduces to the scaled memory-less SR1 method based on sufficient descent or on conjugacy condition. The global convergence of the corresponding algorithm is established under classical assumptions. The numerical results show that the modified memory-less SR1 update based on the cubic regularization is more efficient and more robust than the scaled memory-less SR1 method and than preconditioned BFGS method implemented in CONMIN.

References

1. Al-Baali, M.: Numerical experience with a class of self-scaling quasi-Newton algorithms, Journal of Optimization Theory and Applications, **96**, 533-553 (1998)
2. Andrei, N.: An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. Numerical Algorithms, **42**(1), 63-73 (2006)
3. Andrei, N.: Acceleration of conjugate gradient algorithms for unconstrained optimization. Applied Mathematics and Computation, **213**(2), 361-369 (2009)
4. Andrei, N.: A double parameter scaled BFGS method for unconstrained optimization. Journal of Computational and Applied Mathematics, **332**, 26-44 (2018)
5. Andrei, N.: A double-parameter scaling Broyden–Fletcher–Goldfarb–Shanno method based on minimizing the measure function of Byrd and Nocedal for unconstrained optimization. Journal of Optimization Theory and Applications **178**, 191-218 (2018)
6. Andrei, N.: An adaptive scaled BFGS method for unconstrained optimization. Numerical Algorithms **77**, 413-432 (2018)
7. Andrei, N.: Nonlinear conjugate gradient methods for unconstrained optimization. Springer Optimization and Its Applications 158, (2020)
8. Andrei, N.: Accelerated Scaled Memory-less SR1 method for Unconstrained Optimization. Academy of Romanian Scientists. Technical Report 5/2021 (2021).
9. Aris, R.: The Mathematical Theory of Diffusion and Reaction in Permeable Catalysts, Oxford, UK, (1975)
10. Arzam, M.R., Babaie-Kafaki, S., Ghanbari, R.: An extended Dai-Liao conjugate gradient method with global convergence for nonconvex functions. Glasnik Matematicki, **52**(72), 361-375 (2017)
11. Averick, B.M., Carter, R.G., Moré, J.J., & Xue, G.L.: The MINPACK-2 test problem collection. Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois, Preprint MCS-P153-6092, June (1992)
12. Bebernes, J., Eberly, D.: Mathematical Problems from Combustion Theory. In Applied Mathematical Sciences, **83**, Springer-Verlag, (1989)
13. Bellavia, S., Morini, B.: Strong local convergence properties of adaptive regularized methods for nonlinear least squares. IMA J. Numer. Anal. **35**, (2014)
14. Benson, H.Y., Shanno, D.F.: Interior-point methods for nonconvex nonlinear programming: cubic regularization. Computational Optimization and Applications, **58**(2), 323-346 (2014)
15. Benson, H.Y., Shanno, D.F.: Cubic regularization in symmetric rank-1 quasi-Newton methods. Math. Progr. Comput. **10**, 457-486 (2018)

16. Bianconcini, T., Sciandrone, M.: A cubic regularization algorithm for unconstrained optimization using line search and nonmonotone techniques. Optimization Methods and Software, **31**, 1008-1035 (2016)
17. Bianconcini, T., Liuzzi, G., Morini, B., Sciandrone, M.: On the use of iterative methods in cubic regularization for unconstrained optimization. Computational Optimization and Applications, **60**(1), 35-57 (2013)
18. Biggs, M.C.: Minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and Its Applications, **8**, 315-327 (1971)
19. Biggs, M.C.: A note on minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and Its Applications, **12**, 337-338 (1973)
20. Bongartz, I., Conn, A.R., Gould, N.I.M., Toint, Ph.L.: CUTE: constrained and unconstrained testing environments. ACM Transactions on Mathematical Software, **21**, 123-160 (1995)
21. Broyden, C.G.: Quasi-newton methods and their applications to function minimization. Mathematics of Computation **21**(99), 368-381 (1967)
22. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms. I. General considerations. Journal of the Institute of Mathematics and Its Applications, **6**, 76-90 (1970)
23. Cartis, C., Gould, N.I., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. Math. Program. **130**(2), 295–319 (2011)
24. Cartis, C., Gould, N.I., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. Math. Program. **127**(2), 245–295 (2011)
25. Cartis, C., Gould, N.I., Toint, P.L.: Evaluation complexity of adaptive cubic regularization methods for convex unconstrained optimization. Optim. Methods Softw. **27**(2), 197–219 (2012)
26. Cimatti, G.: On a problem of the theory of lubrication governed by a variational inequality, Applied Mathematics and Optimization **3**, 227–242 (1977)
27. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Convergence of quasi-newton matrices generated by the symmetric rank one update. Mathematical Programming **50**(1-3), 177-195 (1991)
28. Contreras, M., Tapia, R.A.: Sizing the BFGS and DFP updates: A numerical study. Journal of Optimization Theory and Applications, **78**, 93-108 (1993)
29. Davidon, W.C.: Variable metric method for minimization. Research and Development Report ANL-5990. Argonne National Laboratories. (1959)
30. Davidon, W.C.: Variance algorithm for minimization. Computer Journal **10**(4), 406-410 (1968)
31. Davidon, W.C.: Variable metric method for minimization. SIAM J. Optim. **1**(1), 1-17 (1991)
32. Dehmiry, A.H.: The global convergence of the BFGS method under a modified Yuan-Wei-Lu line search technique. Numerical Algorithms, **84**, 781-793 (2020)
33. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming, **91**, 201-213 (2002)

34. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: sequential unconstrained minimization techniques. Research Analysis Corporation, McLean Virginia. (1968) [Republished in 1990 by SIAM, Philadelphia.]
35. Fletcher, R.: A new approach to variable metric algorithms. The Computer Journal, **13**, 317-322 (1970)
36. Fletcher, R., Powell, M.J.D.: A rapidly convergent descent method for minimization, The Computer Journal, 163–168 (1963)
37. Glowinski, R.: Numerical Methods for Nonlinear Variational Problems, Springer-Verlag, Berlin, (1984)
38. Gould, N.I.M., Porcelli, M., Toint, Ph.L.: Updating the regularization parameter in the adaptive cubic regularization algorithm. Computational Optimization and Applications, **53**, 1-22 (2012)
39. Goldfarb, D.: A family of variable metric method derived by variation mean. Mathematics of Computation, **23**, 23-26 (1970)
40. Goodman, J., Kohn, R., Reyna, L.: Numerical study of a relaxed variational problem from optimal design. Computer Methods in Applied Mechanics and Engineering, 57, 107–127 (1986)
41. Griewank, A.: The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical Report NA/12, Department of Applied Mathematics and Theorerical Physics, University of Cambridge. (1981)
42. Griewank, A., Fisher, J., Bosse, T.: Cubic overestimation and secant updating for unconstrained optimization of $C^{2,1}$ functions. Optim. Methods Softw. **29**(5), 1075-1089 (2014)
43. Hsia, Y., Sheu, R.L., Yuan, Y.X.: Theory and application of *p*-regularized subproblems for $p>2$. Optimization Methods & Software, **32**(5), 1059–1077 (2017)
44. Huang, H.Y.: Unified approach to quadratically convergent algorithms for functions minimization. Journal of Optimization Theory and Applications **5**(6), 405-423 (1970)
45. Kelley, C.T., Sachs, E.W.: Local convergence of the symmetric rank-one iteration. Computational Optimization and Applications **9**, 43-63 (1998)
46. Khalfan, H.F., Byrd, R.H., Schnabel, R.B.: A theoretical and experimental study of the symmetric rank-one update. SIAM J. Optimization **3**(1), 1-24 (1993)
47. Liao, A.: Modifying BFGS method. Operations Research Letters, **20**, 171-177 (1997)
48. Nesterov, Y., Polyak, B.T.: Cubic regularization of Newton'smethod and its global performance. Math. Program. **108**(1), 177–205 (2006)
49. Nitsche, J.C.C.: Lectures on Minimal Surfaces. Cambridge University Press, vol. 1, Cambridge, UK (1989)
50. Nocedal, J.: Theory of algorithms for unconstrained optimization. Acta Numerica, **1**, 199-242 (1992)
51. Nocedal, J., Wright, S.J.: Numerical optimization. Springer Series in Operations Research. Springer Science+Business Media, New York, Second edition, (2006)
52. Nocedal, J., Yuan, Y.X.: Analysis of self-scaling quasi-Newton method. Mathematical Programming, **61**, 19-37 (1993)
53. Oren, S.S., Luenberger, D.G.: Self-scaling variable metric (SSVM) algorithms, part I: criteria and sufficient conditions for scaling a class of algorithms. Management Science, **20**, 845-862 (1974)

54. Oren, S.S., Spedicato, E.: Optimal conditioning of self-scaling variable metric algorithm. Mathematical Programming, **10**, 70-90 (1976)
55. Powell, M.J.D.: A new algorithm for unconstrained optimization. In: J.B. Rosen, O.L. Mangasarian, K. Ritter (Eds.) Nonlinear Programming. Academic Press, New York, 31-66 (1970)
56. Shanno, D.F.: Conditioning of quasi-Newton methods for function minimization. Mathematics of Computation, **24**, 647-656 (1970)
57. Shanno, D.F.: Phua, K.H.: Algorithm 500. Minimization of unconstrained multivariable functions. ACM Transactions on Mathematical Software, **2**, 87-94 (1976)
58. Shanno, D.F.: Conjugate gradient methods with inexact searches, Mathematics of Operations Research, **3**, 244-256 (1978)
59. Shanno, D.F.: On the convergence of a new conjugate gradient algorithm, SIAM Journal on Numerical Analysis, **15**, 1247-1257 (1978)
60. Shanno, D.F.: CONMIN – A Fortran subroutine for minimizing an unconstrained nonlinear scalar valued function of a vector variable x either by the BFGS variable metric algorithm or by a Beale restarted conjugate gradient algorithm. Private communication, October 17, 1983.
61. Wolfe, P.: Convergence conditions for ascent methods. SIAM Review, **11**, 226-235 (1969)
62. Wolfe, P.: Convergence conditions for ascent methods. II: Some corrections. SIAM Review, **13**, 185-188 (1971)
63. Yabe, H., Martínez, H.J., Tapia, R.A.: On sizing and shifting the BFGS update within the sized Broyden family of secant updates. SIAM Journal on Optimization, **15**(1), 139-160 (2004)
64. Yuan, Y.X.: A modified BFGS algorithm for unconstrained optimization. IMA Journal of Numerical Analysis, **11**, 325-332 (1991)
65. Yuan, Y.X., Byrd. R.: Non-quasi-Newton updates for unconstrained optimization. Journal of Computational Mathematics, **13**(2), 95-107 (1995)
66. Yuan, G., Sheng, Z., Wang, B., Hu, W., Li, C.: The global convergence of a modified BFGS method for nonconvex functions. Journal of Computational and Applied Mathematics, **327**, 274-294 (2018)
67. Zhao, T., Liu, H., Liu, Z.: New subspace minimization conjugate gradient methods based on regularization model for unconstrained optimization. Numerical Algorithms**,** Optimization online, OO Digest: April 2020, http://www.optimization-online.org/DB_HTML/2020/04/7720.html (2020)

-----ooooo0ooooo-----