# A Symmetric Rank-One Quasi-Newton Line-Search Method Using Negative Curvature Directions

Figen Öztoprak
Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey.
figen@su.sabanciuniv.edu

Ş. İlker Birbil
Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey.
sibirbil@sabanciuniv.edu

Abstract: We propose a quasi-Newton line-search method that uses negative curvature directions for solving unconstrained optimization problems. In this method, the symmetric rank-one (SR1) rule is used to update the Hessian approximation. The SR1 update rule is known to have a good numerical performance; however, it does not guarantee positive definiteness of the updated matrix. We first discuss the details of the proposed algorithm and then concentrate on its numerical efficiency. Our extensive computational study shows the potential of the proposed method from different angles, such as; its second order convergence behavior, its exceeding performance when compared to two other existing packages, and its computation profile illustrating the possible bottlenecks in the execution time. We then conclude the paper with the convergence analysis of the proposed method.

*Keywords*: Quasi-Newton; SR1 update; nonconvexity; negative curvature; unconstrained

**1. Introduction.** Quasi-Newton methods are distinguished by their use of approximate Hessian matrices. These approximate matrices are evaluated with respect to some iterative update formula as the algorithm progresses. The update procedure only requires the gradient of the objective function at each iteration. Thus, these methods provide a way of obtaining some curvature information without evaluating the exact Hessian. This is particularly useful when Hessian is very demanding to compute or cannot be computed at all for some reason. Because they are known to be generally more applicable and quite efficient, quasi-Newton methods are still widely used tools of nonlinear programming even after the development of automatic differentiation packages [20].

There are numerous work on the use of quasi-Newton methods either in line-search or trust-region applications. The methods differ by the formula they use for updating the approximate Hessian matrix [14]. In this paper, we shall focus on the line-search implementation of the symmetric rank-one (SR1) update formula to find an optimal solution of the general unconstrained nonlinear programming problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbb{R}^n, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a real valued differentiable function and its gradient, $\nabla f(.)$, is continuous.

The symmetric rank one update computes the approximate Hessian, $B_{k+1}$ in iteration $k+1$ by using the current approximation, $B_k$ and the gradient of the objective function in two consecutive iterations $\nabla f(x_k)$ and $\nabla f(x_{k+1})$ with

$$B_{k+1} = B_k + \frac{(y_k - B_k v_k)(y_k - B_k v_k)^\intercal}{(y_k - B_k v_k)^\intercal v_k}, \tag{2}$$

where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ and $v_k = x_{k+1} - x_k$. The inverse of the approximate matrices can be calculated in a similar manner by generating an inverse update rule, which can be achieved by applying the Sherman-Morrison-Woodbury formula on the SR1 update rule above [11]. This yields,

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(v_k - B_k^{-1} y_k)(v_k - B_k^{-1} y_k)^\intercal}{(v_k - B_k^{-1} y_k)^\intercal y_k}. \tag{3}$$

The SR1 formula preserves symmetry of the Hessian but the resulting matrix is not necessarily positive definite. This could be a drawback in line search applications, since the corresponding (approximate) Newton direction may fail to be a descent direction. Thus, SR1 formula has generally been used either when the approximations are expected to be positive definite or within the trust-region implementations. Another problem with the SR1 formula is that the denominator of the formula may vanish and cause undefined approximations. A strategy suggested for the solution of this problem is to apply the update formula only if the condition given by

$$|(y_k - B_k v_k)^\intercal v_k| \geq r_1 \|v_k\| \|y_k - B_k v_k\|, \tag{4}$$

is satisfied; otherwise, the update procedure is skipped ($B_{k+1} = B_k$). Applying this strategy, the denominator drawback can be overcome at no significant performance cost because the violation of the above condition is not expected to happen frequently and skipping the update under this condition does not cause a significant loss of curvature information [20]. A similar strategy can be followed to guarantee existence of the inverse matrices,

$$|(v_k - B_k^{-1}y_k)^\intercal y_k| \geq r_2 \|y_k\| \|v_k - B_k^{-1}y_k\|, \tag{5}$$

where $r_1$ and $r_2$ denote small positive real numbers.

The theoretical properties of the SR1 formula have been focus of interest, particularly after the influential work of Conn et al. [8]. In this paper, Conn et al. study the convergence properties of the formula and show 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 [17] provide similar convergence results removing the first of these assumptions. However, Khalfan et al. [18] have observed that the second assumption on uniform linear independence is generally unsatisfied in practice and by removing this assumption, they show $(n+1)$-step superlinear convergence of the method when the approximate matrices are assumed to be positive definite. In a follow-up work, they prove the same convergence result for the case where SR1 updates are used within a trust region framework without the assumption of positive definiteness [7].

Numerical experiments in the literature verify the superior efficiency of the SR1 formula in line search and in trust region applications (for a list of references, see [16]). However, the formula has generally been neglected due to its drawbacks as mentioned above. When the formula is modified to overcome those drawbacks, it becomes less efficient [25]. A solution other than modification of the formula is suggested by [22]. The idea is to switch to the BFGS formula, whenever the SR1 rule is calculated to produce an indefinite matrix.

In this study, we welcome indefinite matrices generated by the SR1 formula and make use of this valuable information about the actual curvature of the objective function. That is, we propose to use the negative curvature directions, whenever the approximate matrices are indefinite. This approach was already used in the literature with the exact Hessian matrices. Mainly, two directions are evaluated; a positive curvature related (modified) Newton direction and a negative curvature direction. Gould et al. [13] propose line search procedures that follow one of these two directions depending on the relative improvement in the objective function. A method, using a combination of these two directions, is proposed by Ferris et al. [10]. In a recent work, Olivares et al. discuss using either a combination or one of these two directions [21]. The global convergence, that is the convergence to the points that satisfy second order necessary conditions, has also been shown for these approaches.

In our view, we make the following contributions:

- ◇ It has been experimentally observed by many researchers that the SR1 rule performs generally better than the other quasi-Newton update rules such as BFGS and DFP. However, SR1 rule is in general not preferred in line search applications due to its risk of failing to find descent directions. Here, we use the SR1 quasi-Newton update rule in a line search context without requiring the positive definiteness of the matrices it generates.
- ◇ The implementation of our approach results in a new algorithm to solve unconstrained problems. We give a thorough computational study and confirm that the numerical performance of the new algorithm is quite promising. This observation is consistent with other observations in the literature, where the SR1 formula has shown superior performance in the trust-region applications using the negative curvature directions as well as the (approximate) Newton directions. We also complement our numerical results with the convergence analysis of the proposed method.
- ◇ The idea of including the negative curvature directions in line search applications has been applied only with exact Hessian matrices before. Our approach adopts a similar idea and uses the approximate Hessian matrices. This not only provides avoiding the calculation of the Hessian matrices but also eliminates the need to factorize the Hessian for inversion at every iteration, since the inverse matrices are readily computed in quasi-Newton applications. We also discuss a way for eliminating the need to compute a negative curvature direction at each step.

The paper is organized as follows. In Section 2, we introduce the proposed algorithm. We present our numerical results to illustrate the performance of the new algorithm in Section 3. We then give in Section 4 the convergence analysis of the proposed algorithm. Finally, we give our conclusions as well as some future research directions in Section 5.

**2. Proposed Algorithm.** We call the proposed algorithm SR1-NC, since it uses the SR1 quasi-Newton update formula and the negative curvature directions. The algorithm applies the main phases of a typical line-search procedure as shown in Algorithm 1. The first two lines of Algorithm 1 define the initialization phase (lines 1-2), where $x_0$, $B_0$, $k_{\max}$ denote the initial iterate, the initial approximate matrix and the maximum number of iterations, respectively.

In the direction computation phase, the usual quasi-Newton direction $s_k$ is computed (line 4). We also compute a negative curvature direction $d_k$ *only if* we suspect that the $B_k$ matrix is indefinite (lines 5-7, where $\lambda_{\min}(B_k)$ denotes the smallest eigenvalue of $B_k$ and $w_{\min}$ is the corresponding eigenvector). That is, we consider the negative curvature direction in the following two cases:

i. $y_k^\intercal v_k < 0$, the update formula indicates negative curvature;

ii. $s_k^\intercal \nabla f(x_k) \geq 0$, the usual quasi-Newton direction is not a descent direction.

When one of these conditions hold, we apply eigenvalue decomposition and set $d_k$ collinear to $w_{\min}$. At each iteration, either the direction $s_k$ or the direction $d_k$ is followed. If both directions are computed, then the estimated decreases they provide are compared. Since we cannot guarantee that the matrices $B_k$ include the correct curvature information, we may not successfully identify the saddle points. Thus, we stop the algorithm when the norm of the gradient function value is close to zero (line 3). By the same reason, we put the safeguard in line 13 and ensure that we have a descent direction unless the termination condition is satisfied. It is important to note at this point that in all numerical tests, we observed that the condition in line 13 occurred quite rarely; in fact, this condition is used only for a few higher dimensional problems that are solved with the large-scale implementation. We shall emphasize about this remark again in the computational results section, where we point out the strong empirical support for the second order convergent behavior of the proposed algorithm.

After the direction computation, the algorithm applies an adaptive line-search, which is very similar to the one in [13]. The line-search procedure is based on either the linear approximation or the quadratic approximation of $f(x_k + \alpha d_k)$. In our implementation of the algorithm, we calculated the step size $\alpha_k$ by applying a backtracking procedure that is common in Newton-type methods. When $s_k$ is selected, we start with an initial trial step size of 1 and reduce it until the condition given by relation (6) is satisfied. On the other hand, when $d_k$ is selected, we start with the step size value that had been used the last time again with the negative curvature direction. If the condition given by relation (6) is violated, we reduce that value until this condition is satisfied. If the initial trial step size satisfies this condition, we increase that value until we get the largest step size that does not violate the condition.

Finally, the new iterate is computed by using the selected direction and the step size (line 18). Right before the end of one iteration, the approximate matrices are updated with the new curvature information (lines 19-24). To avoid generating undefined matrices, we update $B_k$ and its inverse only when the conditions given by relations (4) and (5) are satisfied.

**3. Computational Study.** To evaluate the performance of Algorithm 1, we have conducted two sets of experiments with small-scale and large-scale implementations of the algorithm. In our subsequent discussion, we refer to the small-scale implementation as SR1-NC and the large-scale implementation as LSR1-NC. All results have been taken on a machine with 2.0 GB RAM and a 2.20 GHz dual core processor. The algorithm has been coded in C++ and compiled with Intel C++ compiler v.11 under Ubuntu 7.10 operating system. We have used the double precision BLAS and LAPACK [2] procedures in Intel Math Kernel Library for all linear algebra operations. The source code of the program as well as the details of the test results given in this paper can be downloaded from [3].

We have used the well-known CUTEr problem set [12]. The most recent version of this set is obtained from [1]. We have compiled two sets of test problems. In the first set, there are 81 small-scale test problems, which involve all the unconstrained CUTEr problems with at most 200 variables. In the second set there are additional 60 large-scale problems and the dimensions of these problems vary between 500 to 10,000. While conducting our experiments, we did not alter the default parameter values of any one of the test problems. Similarly, the initial solution point is kept as the default one provided by the CUTEr package.

**3.1 Tests with SR1-NC.** For the first set of experiments, we have selected two packages as benchmarks, UNCMIN [23] and TENMIN [24], both of which have been designed to solve small-to-medium size unconstrained problems. Moreover, both packages have an interface to CUTEr. These packages are coded in FORTRAN77 language. The UNCMIN and TENMIN packages are obtained from [5] and [4], respectively.

**Algorithm 1**: SR1-NC

**Input**: $x_0, \mu \in (0,1), \tau > 1, k_{\max}, B_0, B_0^{-1}, \epsilon_P > 0, \epsilon_M > 0$
$y_0 = 0,\ v_0 = 0,\ k = 0$
**while** $k < k_{\max}$ **and** $\|\nabla f(x_k)\| > \epsilon_P$ **do**
  $s_k = -B_k^{-1} \nabla f(x_k)$
  **if** $y_k^\intercal v_k < 0$ **or** $s_k^\intercal \nabla f(x_k) \geq 0$ **then**
    Apply eigenvalue decomposition to solve $B_k w_{\min} = \lambda_{\min}(B_k) w_{\min}$
    $d_k = -\mathrm{sgn}(w_{\min}^\intercal \nabla f(x_k)) \frac{w_{\min}}{\|w_{\min}\|}$
  **else**
    $d_k = 0$
  **if** $s_k^\intercal \nabla f(x_k) \leq \tau \|s_k\| (d_k^\intercal \nabla f(x_k) + \frac{1}{2} d_k^\intercal B_k d_k)$ **then**
    $p_k = s_k$
  **else**
    **if** $|d_k^\intercal \nabla f(x_k)| \leq \epsilon_M ||\nabla f(x_k)||$ **then**
      $p_k = -\nabla f(x_k)$
    **else**
      $p_k = d_k$
  Compute a step length $\alpha_k > 0$ such that

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu[\alpha_k \nabla f(x_k)^\intercal p_k + \frac{1}{2}\alpha_k^2 \min(0, p_k^\intercal B_k p_k)]. \quad (6)$$

  $x_{k+1} = x_k + \alpha_k p_k$
  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$
  $v_k = x_{k+1} - x_k$
  **if** (4) **and** (5) *are satisfied* **then**
    Compute $B_{k+1}$ and $B_{k+1}^{-1}$ using (2) and (3), respectively
  **else**
    $B_{k+1} = B_k$ and $B_{k+1}^{-1} = B_k^{-1}$
  $k = k + 1$

In all our tests, we have used the parameters listed in Table 1 for all packages. With SR1-NC, we have used the following algorithm specific parameters: $B_0 = I$, $\mu = 10^{-3}$, $\epsilon_M = 0$ and $\tau = 2.0$. To terminate any one of the packages, we have used the following five exit codes[1]: (1) the gradient is close to zero, (2) step size is close to zero, (3) there is no descent direction, (4) maximum number of iterations is exceeded, and (5) maximum step size is exceeded in five consecutive iterations. We report all our results with UNCMIN and TENMIN in Table 3 and Table 4, and with SR1-NC in Table 5 and Table 6 of Appendix A.

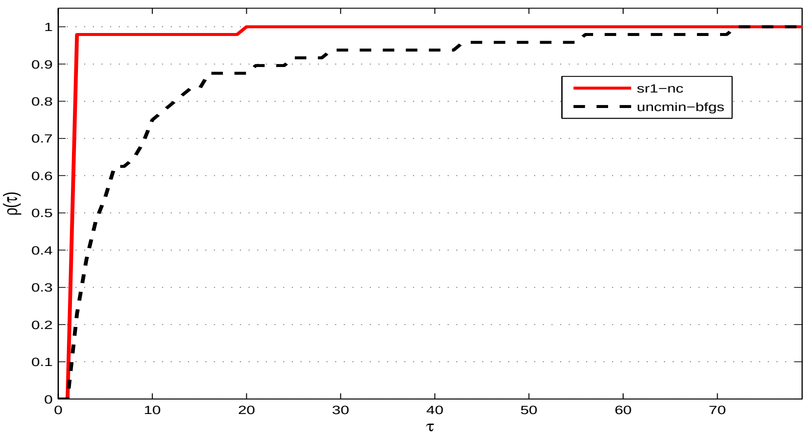
**3.1.1 UNCMIN and TENMIN Benchmarks.** We first compare our results with that of the well-known BFGS quasi-Newton update rule. The results for BFGS are obtained running the UNCMIN package with the proper options. It is important to note that for fair comparison we only include the 48 cases, for which both UNCMIN and SR1-NC have converged to the same point. The problems included in the benchmark are marked with a † sign in Table 3 and Table 4. We did not use CPU times to compare the performances because UNCMIN has been coded in Fortran. We observe that C++ requires more CPU time than Fortran. There are, however, a number of work comparing the relevant compilers and arguing that the running time performance of these two programming languages would be comparable after certain fine-tuning [15, 26]. Since the computation times for all problems are quite small and the differences are negligible, we did not dwell on such fine-tunings within the scope of this work.

[1] We modified the exit codes for SR1-NC to comply with the ones given for both UNCMIN and TENMIN.

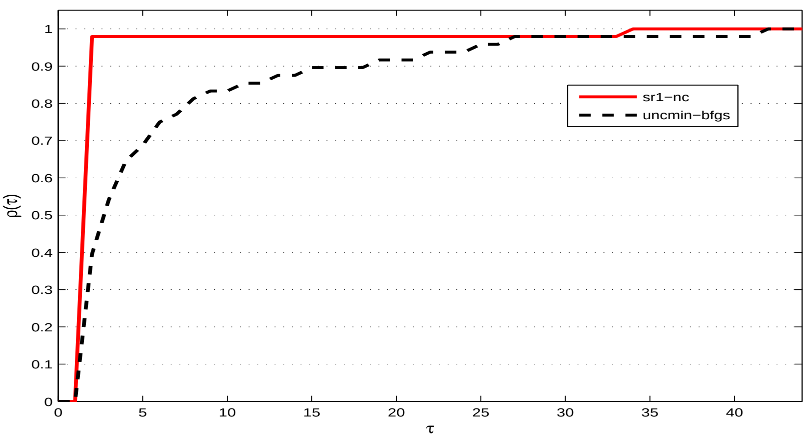
Table 1: The general parameters used in all tests.

| | |
|---|---|
| Maximum number of iterations: | $100 \times n$ |
| Gradient tolerance: | $10^{-6}$ |
| Stepsize tolerance: | $10^{-8}$ |
| Maximum stepsize: | $1,000 \times \max\{\|x_0\|_\infty, 1.0\}$ |
| Scale parameter: | 1.0 |

Figure 1 shows the performance profiles for the number of function and gradient evaluations (see [9] for the details about performance profiles). Clearly, when UNCMIN uses the BFGS update rule (UNCMIN-BFGS), SR1-NC outperforms UNCMIN-BFGS in both the number of function and the number of gradient evaluations.




(a) Number of function evaluations.




(b) Number of gradient evaluations.

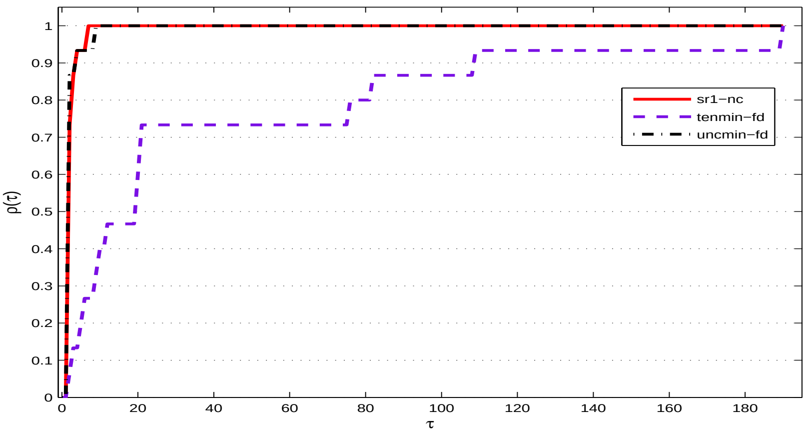
Figure 1: Performance profiles for SR1-NC (Part I).

We also compare our results with that of the UNCMIN package when finite difference Hessian approximations are used (UNCMIN-FD), and with that of the TENMIN package when the tensor method applied with finite difference Hessian approximations (TENMIN-FD). Consequently, we exhaust all options of UNCMIN and TENMIN for solving small-to-medium size problems using line-search without the exact Hessian matrices. Table 2 displays the number of problems that each solver could solve successfully for different levels of accuracy ($\|\nabla f(x)\|_\infty \leq \epsilon_P$). Notice that the performance of SR1-NC does not deteriorate as the precision increases. This is an indication about the robustness of the proposed algorithm.

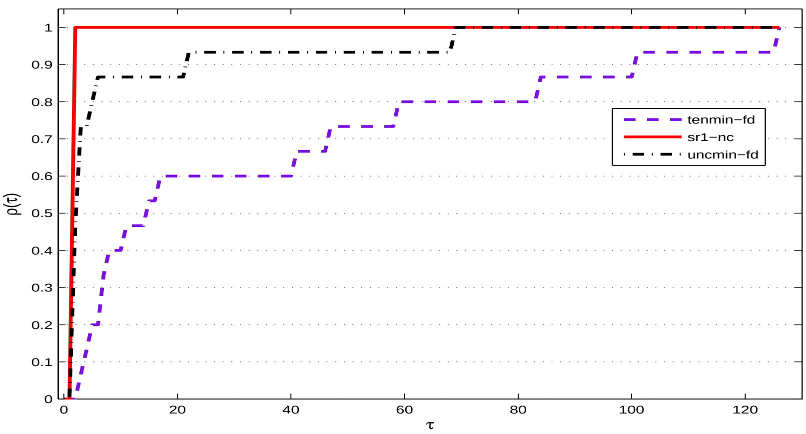
In Figure 2, the performance profiles are provided for the number of function and gradient evaluations. The finite difference approximation implementations generally did not converge to the same points as SR1-NC. Therefore, the performance profiles could be obtained over a restricted set of only 15 problems, which are marked with the sign ‡ in Table 3 and Table 4. The results show that the performance of SR1-NC is compatible in terms of number of function evaluations and superior in terms of number of gradient evaluations, when compared with finite difference approximations.

Table 2: Number of problems solved successfully within the given iteration limit.

| Package | $\epsilon_P = 10^{-2}$ | $\epsilon_P = 10^{-4}$ | $\epsilon_P = 10^{-6}$ |
|---|---|---|---|
| UNCMIN-BFGS | 51 | 45 | 32 |
| UNCMIN-FD | 67 | 59 | 54 |
| TENMIN-FD | 28 | 22 | 14 |
| SR1-NC | 63 | 60 | 55 |




(a) Number of function evaluations.




(b) Number of gradient evaluations.

Figure 2: Performance profiles for SR1-NC (Part II).

Both benchmark results above are quite promising for SR1-NC. Figures 1 and 2 show that for around 98% of the problems in our test set, SR1-NC was in general more efficient than the other two algorithms in terms of function and gradient evaluations. There is, however, one exception in terms of the number of function evaluations. As shown by Figure 2(a) UNCMIN package with finite difference approximations shows a performance that is quite close to SR1-NC. One remaining question is the cost of eigenvalue decomposition, which we did not involve in the benchmark above. This issue will be revisited later.

**3.1.2 The Convergence Performance.** In this part, we try to analyze the performance of Algorithm 1 by considering the effects of different factors. This analysis includes 60 instances for which our algorithm could obtain a final gradient with an infinity norm less than $10^{-5}$. The corresponding problems are marked with a ‡ sign in Table 5 and Table 6.

We have mentioned that in Algorithm 1 the update of the approximate Hessian $B_k$ is skipped when conditions (4) or (5) are violated. However, in our computational study, these conditions are almost always satisfied. In fact, over *all iterations* executed during our tests with 81 problems, such a case occurred in only 4 iterations. On the other hand, for all problems the condition $d_k^\intercal \nabla f(x_k) \leq \epsilon_M$ given in line 13 of Algorithm 1 has never occurred at any iteration.

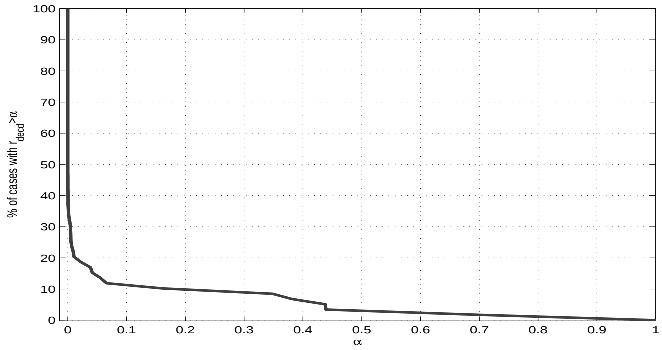
We also observed that the number of cases terminated with an exit code of 2 was remarkable (see column 4 of Table 5 and Table 6). So, we also implemented our algorithm with a stepsize tolerance of zero, i.e., no terminations are allowed due to very tiny steps. There were 64 successful instances in this case, and only 4 problems have been prematurely terminated because of the inappropriate stepsize tolerance parameter.

The first question that we had is related to the global convergence performance of the proposed algorithm. To check whether the algorithm has exactly converged to a local minimum point, we calculated the exact Hessian of the objective function at the final iteration for each problem. As it can be seen from column 10 of both Table 5 and Table 6, except 6 instances out of 60 successful cases, the algorithm has converged to a point where the exact Hessian is positive definite. This demonstrates empirically that the proposed algorithm is capable of converging to points that satisfy second order sufficient conditions. We have also calculated the Frobenious norm of the relative Hessian approximation error at the final point. The corresponding error values are given in column 9 of both Table 5 and Table 6.

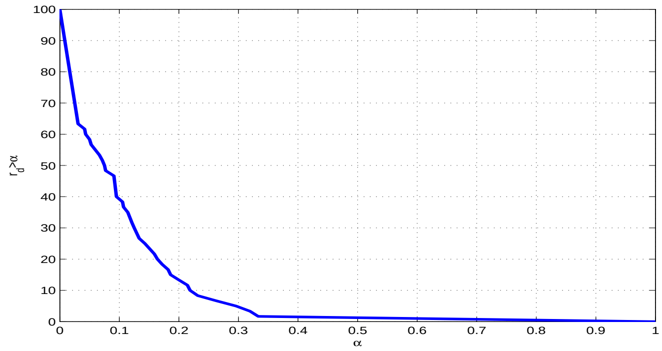
The main motivation of using negative curvature directions is that it prevents the standard line-search algorithm from failing when indefinite approximate Hessian matrices are obtained. But does it also provide further performance advantages? How does it contribute to the nice performance results of the new algorithm? How is the improved performance related to the negative curvature directions as well as the SR1 update rule itself? To answer these questions and understand the effect of negative curvature directions, we have decided to control the success of steps taken by the negative curvature directions. Figure 3(a) shows the percentage of cases among all problem instances for which the ratio of the decrease in the objective function value, achieved by selecting the negative curvature directions over total objective function value decrease, is above a given threshold ($r_{decd} > \alpha$). For instance the point (0.2, 10) on the graph implies that among 10% of all problems, at least 20% of the total decrease in the objective function value is obtained due to the selection of negative curvature directions (see also column 13 of Table 5 and Table 6). Similarly, Figure 3(b) illustrates the percentage of cases for which the ratio of the number of iterations, at which the negative curvature direction has been followed, over total number of iterations is above a threshold ($r_d > \alpha$) (see also column 12 of Table 5 and Table 6). We observe that for 40% of all problems, at least 1 in every 10 iterations is carried out by following the negative curvature directions. All these results indicate that although the negative curvature direction steps are not the primary steps of the proposed algorithm, they provide the crucial information when the algorithm has got stuck. In addition, one can also concur that the negative curvature direction steps have been complementary to the gradient related steps rather than being alternatives.

**3.1.3 Cost of Backtracking and Decomposition.** We also observe that the number of function evaluations per iteration could be high for SR1-NC (see column 15 of Table 5 and Table 6 in Appendix A). In Figure 4, we illustrate the cost of backtracking line-search for all instances by plotting the percentage of cases for which the ratio of the number of function evaluations during line-search over total number of iterations is above a threshold ($r_{fb} > \alpha$). The figure supports our hypothesis about the bottleneck caused by backtracking line-search; for 60% of all problems, at least half of total number of iterations is spent in the backtracking line-search procedure.

In Algorithm 1 we do not calculate the minimum eigenvalue at each iteration, but eigenvalue decomposition is still another costly operation of the algorithm. To get an insight about this cost, for each problem instance we measured the time spent for decomposition and calculated its ratio over the total solution time ($r_{dt}$). We note at this point that for those test problems for which the total solution time is negligibly small, we set $r_{dt} = 0$. The results given in Figure 5 show that for only 10% of all problems, the time spent for decomposition exceeds approximately 15% of total solution time (see also column 14 of Table 5 and Table 6). The smallest eigenvalue and the corresponding eigenvector can also be calculated approximately in a less costly way for the large-scale implementation of the algorithm. This issue shall




(a) Objective function value decrease provided by negative curvature directions.




(b) Frequency of using negative curvature directions.

Figure 3: Performance effect of negative curvature directions in SR1-NC.

be further argued in the next section.

**3.2 Tests with LSR1-NC.** In the large-scale implementation of the algorithm, we have used a limited memory SR1 update routine together with the Lanczos procedure. We have applied the compact form formulations of the limited memory updates [20].

We have compared our results with that of the well-known L-BFGS method. The computer program for L-BFGS is obtained from [19]. To have a fair comparison, we have adopted the termination condition of L-BFGS [2] and modified the exit codes of LSR1-NC: 0, successful termination; -1, violation of stepsize bounds, i.e., $\alpha_k \leq \alpha_{\min}$ or $\alpha_k \geq \alpha_{max}$; 1, the maximum number of iterations is exceeded. The parameters used in LSR1-NC are set to the default values in L-BFGS: $\epsilon_P = 10^{-5}$; stepsize tolerance, $\alpha_{\min} = 10^{-20}$; maximum stepsize, $\alpha_{\max} = 10^{20}$; maximum number of iterations, 10.000; the size of the memory, 5 pairs. For computing the minimum eigenvalue and the corresponding eigenvector approximately, we have applied the Lanczos procedure [11]. In our implementation, we have limited the effort spent for negative curvature direction computation by setting the maximum number of orthogonal base vectors computed by the procedure to $\min(n, 20)$.

We first present the benchmark results. The unconstrained problems of the CUTEr set with at most 10,000 variables are solved with both L-BFGS and LSR1-NC. The problem set contained a total of 141 test problems. The complete results for both algorithms are given in Tables 7 and Table 9 of Appendix A. In 14 instances, both algorithms terminated with an exit code of either 1 or $-1$. There are 12 instances for which LSR1-NC terminated successfully but L-BFGS either exceeded the maximum number of iterations or ended up with a line-search fail. On the other hand, for 5 instances, L-BFGS terminated successfully but LSR1-NC either exceeded the number of iterations or stopped because of a line-search error.

We have conducted a benchmark over 91 instances, for which the difference between the final objective function values reported by both algorithms is less than $10^{-3}$. Figure 6 shows the performance profiles in terms of number of function and gradient calls. In terms of number of function calls, L-BFGS outperforms

[2] $||\nabla f(x_k)||/\max(1, ||x_k||) < \epsilon_P$

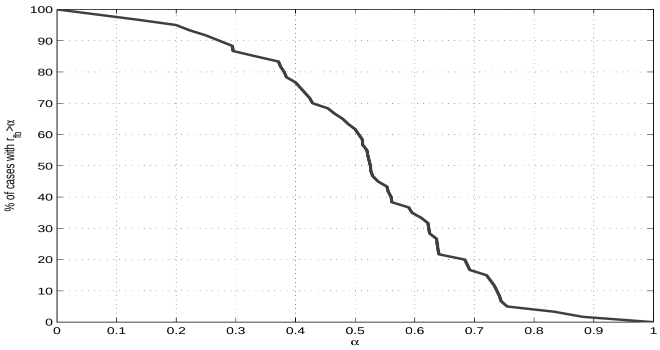



Figure 4: Cost of backtracking line-search in SR1-NC.

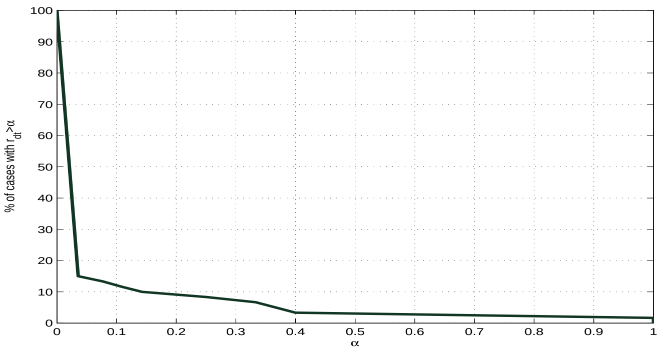


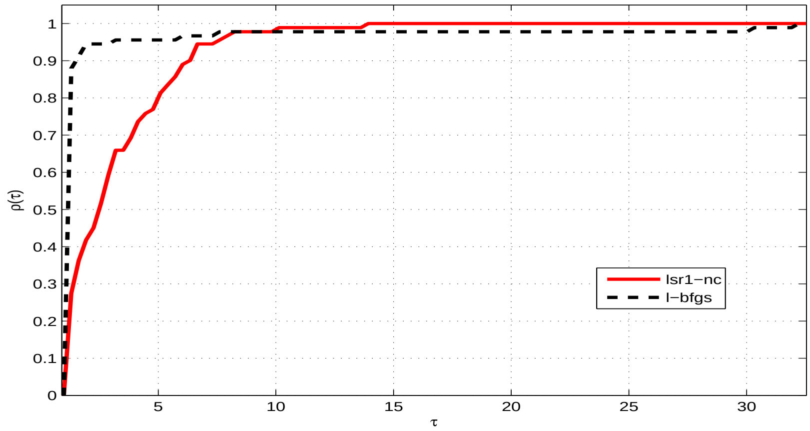

Figure 5: Cost of eigenvalue decomposition in SR1-NC.

LSR1-NC (see Figure 6(a)). This may be again credited to the backtracking line-search procedure (see also our subsequent discussion about Figure 8). Nonetheless, in terms of number of gradient evaluations, LSR1-NC performs better than L-BFGS as shown in Figure 6(b).

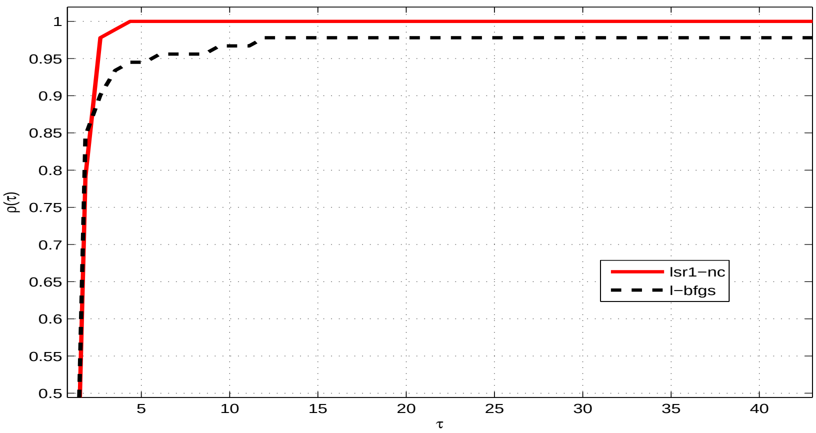
To get more insight about the performance of the large-scale implementation, we repeat the analysis we have done for the small-scale version. Figure 7(a) to Figure 9 are the large-scale counterparts of Figure 3(a) to Figure 5, respectively. Figure 7(a) and Figure 7(b) show similar patterns as in the small-scale implementation. On average, we observe a decrease in the frequency of using negative curvature directions. Clearly, the most significant difference is in the cost of decomposition, which is much lower for LSR1-NC thanks to the Lanczos procedure (see Figure 9). However, as illustrated in Figure 8, the backtracking line-search procedure in the large-scale implementation uses up a larger portion of the total number of function evaluations than the small-scale implementation.

Finally, to test the convergence behavior of LSR1-NC, we have also checked whether the exact Hessian matrix at the final solution point of the algorithm is positive definite. Note that, we have restricted the size of the memory to only 5 pairs for Lanczos procedure, even though the problem sizes scale up to $10,000$ dimensions. Therefore, initially we did not expect to see a very successful final curvature approximation. To our surprise, the results have shown that in only 12 cases out of 122 successfully solved instances, the Hessian matrix at the final solution point is not positive definite, and for other 7 cases the eigenvalue decomposition procedure has failed. We have also tested whether the orthogonality condition (line 13 of Algorithm 1) has ever occurred in the large-scale implementation. We have observed that out of 141 problem instances, this condition has been activated only for 15 instances.

**4. Convergence Analysis.** We devote this section to the convergence analysis of Algorithm 1. Basically, we first discuss that the algorithm is well-defined and then show that it is first order convergent. Since we compute the curvature approximately, unless we have some information about the exact curvature of the function or how well it is approximated, we do not have any choice but to base some parts of our convergence proof on the gradient information. As it shall be clear from the proof of the next lemma,




(a) Number of function evaluations.




(b) Number of gradient evaluations.

Figure 6: Performance profiles for LSR1-NC.

we needed the extra step in line 13 of Algorithm 1 to make sure that there exists at least one gradient related direction at each step.

Here are two standard assumptions that we use in our subsequent results:

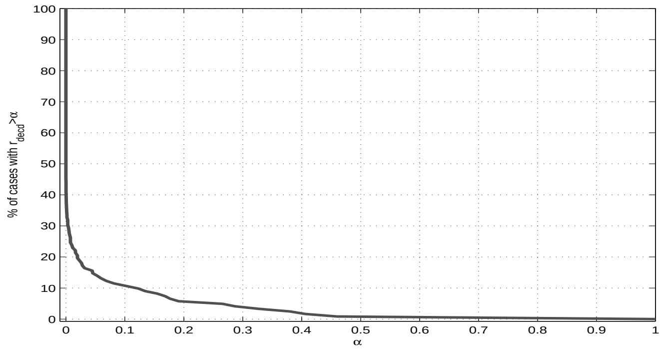
A1. The matrices $B_k$ and $B_k^{-1}$, $k \in \mathbb{N}$ are bounded.

A2. The function $f(\cdot)$ is bounded below and the lower level set of $x_0$ is compact.

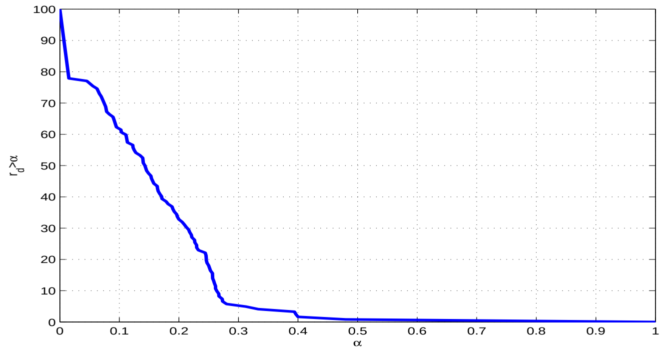
LEMMA 4.1 *Suppose assumption A1 holds. Then, at a nonstationary point $x_k$, Algorithm 1 computes at least one nonzero direction vector, $p_k \neq 0$ satisfying*

$$\begin{array}{ll} (i) & p_k^{\intercal}\nabla f(x_k) < \kappa||\nabla f(x_k)|| \\ (ii) & \|p_k\| \leq M \end{array} \tag{7}$$

*for some $\kappa > 0$, $M > 0$.*




(a) Objective function value decrease provided by negative curvature directions.




(b) Frequency of using negative curvature directions.

Figure 7: Performance effect of negative curvature directions in LSR1-NC.

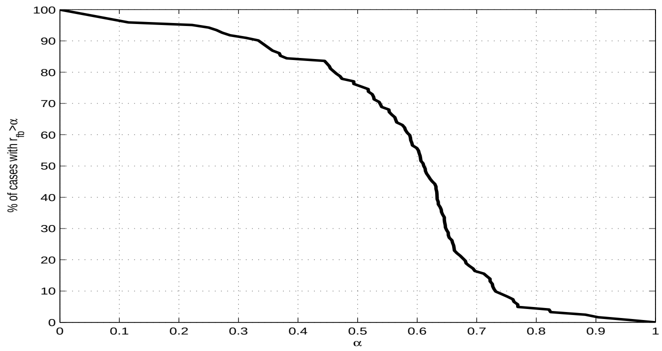



Figure 8: Cost of backtracking line-search in LSR1-NC.

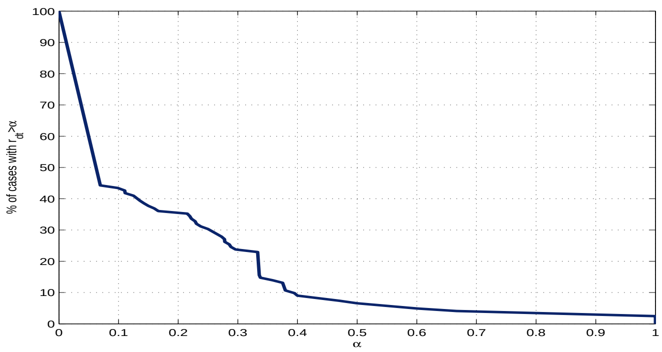



Figure 9: Cost of Lancsoz procedure in LSR1-NC.

PROOF.

I. Suppose $B_k$ is positive definite. Then, the conditions in line 5 of Algorithm 1 are not satisfied, and hence, $d_k = 0$. Thus, we have

$$p_k = s_k = -B_k^{-1}\nabla f(x_k) \neq 0.$$

Since

$$\|s_k\| = \|B_k^{-1}\nabla f(x_k)\| \leq \|B_k^{-1}\|\|\nabla f(x_k)\|$$

and

$$s_k^\intercal \nabla f(x_k) = -\nabla f(x_k)^\intercal B_k^{-\intercal}\nabla f(x_k) \leq \|B_k^{-1}\|\|\nabla f(x_k)\|^2,$$

conditions (i) and (ii) are satisfied by using the boundedness of the approximate matrices.

II. Suppose $B_k$ is indefinite. Then there are two cases:

a. If $s_k^\intercal B_k s_k < 0$, then $s_k^\intercal \nabla f(x_k) > 0$. This implies that

$$d_k = -\text{sgn}(w_k^\intercal \nabla f(x_k))\frac{w_k}{\|w_k\|} \neq 0$$

with $B_k w_k = \lambda_{\min}(B_k)w_k$. Note that $d_k^\intercal \nabla f(x_k) < 0$ and

$$(d_k)^\intercal B_k d_k = \frac{1}{\|w_k\|^2}(w_k)^\intercal B_k w_k = \lambda_{\min}(B_k) < 0.$$

Thus, the condition in line 10 of Algorithm 1 is not satisfied. Furthermore, if $|d_k^\intercal \nabla f(x_k)| > \epsilon_M||\nabla f(x_k)||$, then $p_k = d_k$ satisfies (i). Since $\|d_k\| = 1$, (ii) is trivially true. Otherwise, if $|d_k^\intercal \nabla f(x_k)| \leq \epsilon_M||\nabla f(x_k)||$, then $p_k = -\nabla f(x_k)$, which clearly satisfies both (i) and (ii).

b. If $s_k B_k s_k > 0$, then selecting $p_k = s_k$ or $p_k = d_k$ works by the previous arguments in part I and part II.a, respectively.

□

We next prove the existence of a positive step length at each iteration of the algorithm.

LEMMA 4.2 *Let $p_k \in \mathbb{R}^n$ be the direction vector selected by Algorithm 1 at iteration $k$. Then, there exists a step length $\alpha_k > 0$ such that (6) holds.*

PROOF. By Lemma 4.1, at each iteration $k$, Algorithm 1 selects a nonzero direction $p_k$ satisfying condition (7). Suppose for contradiction that there exists no $\alpha_k$ satisfying (6). Then, there exists a sequence $\alpha_j \downarrow 0$ as $j \uparrow \infty$ such that

$$f(x_k + \alpha_j p_k) - f(x_k) > \mu[\alpha_j \nabla f(x_k)^\intercal p_k + \frac{1}{2}\alpha_j^2 \min(0, p_k^\intercal B_k p_k)].$$

By using the mean value theorem and dividing both sides by $\alpha_j$, we get for $\theta \in (0, 1)$

$$\nabla f(x_k + \theta\alpha_j p_k)^\intercal p_k > \mu \nabla f(x_k)^\intercal p_k + \mu\frac{1}{2}\alpha_j \min(0, p_k^\intercal B_k p_k).$$

However, for $j \uparrow \infty$ we obtain $(1-\mu)\nabla f(x_k)^\intercal p_k > 0$, which contradicts that $\nabla f(x_k)^\intercal p_k < 0$. □

We shall next conclude in Theorem 4.1 that Algorithm 1 converges to a point that satisfies the first order conditions.

THEOREM 4.1 *Suppose assumptions A1 and A2 hold. Then, the algorithm converges to a point $x_*$ with $\|\nabla f(x_*)\| = 0$.*

PROOF. Note that

$$f(x_k + \alpha_k p_k) \leq f(x_k) + \mu[\alpha_k \nabla f(x_k)^T p_k + \frac{1}{2}\alpha_k^2 \min(0, p_k^T B_k p_k)] \leq f(x_k) + \mu\alpha_k \nabla f(x_k)^T p_k,$$

and $p_k$ is a gradient related direction by Lemma 4.1. Using also Lemma 4.2, we can apply the standard convergence analysis for Armijo line-search (see, for instance, [6]).

□

A very impressive property of the SR1 update is that the approximate matrices it produces are expected to get closer to the exact Hessian. Its limiting convergence behavior, $\|B_k - \nabla^2 f(x_*)\| \downarrow 0$, has already been studied in the literature [8]. Such a result requires the assumption of uniform linear independence of the search directions as well as the boundedness and Lipschitz continuity of $\nabla^2 f(x_k)$. In our numerical tests, we have observed that the proposed algorithm has converged to the points that satisfy second order necessary conditions for most of the successfully solved test problems. This is also true for large-scale implementation, even in that case we could not extensively use the approximate curvature information produced by the SR1 updates. On the other hand, it is relatively simple to discuss that the algorithm may fail to converge to minimizers in the most general case. For example, if the initial Hessian approximation is selected as a positive definite one and the initial iterate turns out to be a saddle point of the objective function, the algorithm terminates at this initial point, which is clearly not a minimizer.

**5. Conclusion and Future Research.** In this paper, we have proposed a quasi-Newton algorithm, SR1-NC, which uses the efficient symmetric rank-one (SR1) update rule with problems that include local nonconvexity. We also implemented a large scale adaptation of the proposed algorithm and evaluated its performance with a thorough computational study on a set of well-known test problems.

Our numerical experiments have revealed that the performance of SR1-NC is quite promising for solving unconstrained nonlinear programming problems. Although the negative curvature information is not used very frequently, we have observed that it plays a crucial role whenever a conventional SR1 quasi-Newton implementation gets stuck. The proposed algorithm requires eigenvalue decomposition, which could affect the numerical performance. For this reason, we have avoided applying an eigenvalue decomposition operation at each step. We have proposed a strategy to apply this operation whenever it is necessary. The numerical results have confirmed the success of this strategy: The number of decompositions was quite few when compared against the number of iterations. Along the same line, we also observed that there were no unnecessary decompositions meaning that the negative curvature direction was always used when it was computed (see columns 11 and 12 of Table 5 and Table 6). For our small-to-medium size test problems, we did not observe any performance problems with respect to the computation of the exact eigenvalues and eigenvectors. However, such factorization could be a clear obstacle for solving large-scale problems. Therefore, we have consulted the well-known Lanczos procedure to estimate the smallest eigenvalue. This approach, as implemented in the large-scale version of the proposed algorithm (LSR1-NC), turned out to be quite successful. Our empirical study suggested that the overall performance of LSR1-NC could be further improved, if one can substitute an effective line-search method for the backtracking procedure.

**Acknowledgments.** We would like to thank Professor Ph. L. Toint and Professor Y.-X. Yuan for their invaluable suggestions and feedback on an earlier version of this work that we presented in Veszprém Optimization Conference: Advanced Algorithms (VOCAL), Veszprém, Hungary, 2008. We also would like to thank Professor J. Burkardt, who kindly helped us with the double precision code of UNCMIN package. We are grateful to Professor A. R. Conn for his careful reading and constructive comments. Finally, we would like to acknowledge The Scientific and Technological Research Council of Turkey (TÜBİTAK) for their support under grant 107M455.

**Appendix A. Computational Results.** We give our test results with UNCMIN, TENMIN and SR1-NC in this section. The list of abbreviations that are used in the tables are as follows:

(1) $n$: problem dimension

(2) $E$: exit code

(3) $N_f$: number of function evaluations

(4) $N_{\nabla f}$: number of gradient evaluations

(5) $||\nabla f_*||_\infty$: infinity norm of the final gradient

(6) $F$: objective function value at the final point

(7) $Err_H$: Hessian error at the final point

(8) $PD$: positive definiteness of the final Hessian (1/0) ; (?) indicates eigenvalue decomposition has failed

(9) $N_{ED}$: number of eigenvalue decompositions

(10) $N_d$: number of iterations in which a negative curvature direction has been followed

(11) $r_{decd}$: ratio of the objective function value decrease by negative curvature directions

(12) $r_{dt}$: ratio of the CPU time spent for eigenvalue decomposition

(13) $N_{fb}$: number of function evaluations in backtracking line-search

Table 3: The results obtained with UNCMIN and TENMIN.

| | | | TENMIN | | | | | UNCMIN-BFGS | | | | | UNCMIN-FD | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *no* | *prob* | $n^{(1)}$ | $E^{(2)}$ | $N_f^{(3)}$ | $N_{\nabla f}^{(4)}$ | $\|\|\nabla f_*\|\|_\infty^{(5)}$ | $F^{(6)}$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | $F$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | $F$ |
| 1 | AKIVA | 2 | 4 | 1122 | 603 | 5.28E-01 | 6.18 | 5 | 9 | 7 | nan | NAN | 1 | 9 | 19 | 1.65E-09 | 6.17 |
| 2 | ALLINITU†‡ | 4 | 1 | 310 | 465 | 3.35E-06 | 5.74 | 1 | 87 | 31 | 1.50E-07 | 5.74 | 1 | 20 | 41 | 5.20E-10 | 5.74 |
| 3 | ARGLINA | 200 | 1 | 202 | 202 | 1.81E-04 | 200 | 1 | 203 | 2 | 2.62E-15 | 200 | 1 | 202 | 202 | 1.81E-04 | 200 |
| 4 | ARGLINB† | 200 | 2 | 204 | 603 | 5.15E+13 | 7.74E+14 | 3 | 277 | 12 | 1.20E+05 | 99.63 | 2 | 207 | 1207 | 1.67E+03 | 99.63 |
| 5 | ARGLINC† | 200 | 2 | 204 | 603 | 5.00E+13 | 7.47E+14 | 3 | 302 | 15 | 7.05E+01 | 101.13 | 3 | 206 | 1006 | 1.59E-01 | 101.13 |
| 6 | BARD†‡ | 3 | 2 | 209 | 152 | 8.49E-05 | 0.01 | 1 | 167 | 50 | 2.56E-07 | 0.01 | 1 | 11 | 29 | 2.59E-08 | 0.01 |
| 7 | BEALE†‡ | 2 | 4 | 828 | 603 | 2.81E-02 | 0 | 1 | 31 | 16 | 1.73E-07 | 1.32E-014 | 1 | 11 | 22 | 5.20E-08 | 2.76E-15 |
| 8 | BIGGS6† | 6 | 2 | 192 | 455 | 1.54E-01 | 0.3 | 1 | 542 | 157 | 5.03E-08 | 0.01 | 4 | 608 | 4201 | 8.25E-03 | 0.03 |
| 9 | BOX3† | 3 | 4 | 1497 | 1204 | 1.85E-04 | 0.07 | 1 | 199 | 48 | 1.97E-07 | 3.89E-014 | 1 | 12 | 33 | 1.01E-10 | 5.67E-19 |
| 10 | BRKMCC†‡ | 2 | 4 | 1137 | 603 | 2.57E-05 | 0.17 | 1 | 81 | 19 | 4.58E-07 | 0.17 | 1 | 6 | 10 | 5.53E-13 | 0.17 |
| 11 | BROWNAL† | 200 | 2 | 205 | 603 | 1.26E+00 | 0.18 | 2 | 272 | 49 | 2.37E-06 | 1.44E-009 | 1 | 348 | 29347 | 2.76E-10 | 1.01E-20 |
| 12 | BROWNBS | 2 | 2 | 126 | 60 | 3.82E+05 | 3.65E+10 | 5 | 4 | 2 | 2.00E+06 | 9.98E+011 | 5 | 4 | 4 | 2.00E+06 | 9.98E+11 |
| 13 | BROWNDEN†‡ | 4 | 4 | 1302 | 2005 | 6.73E-01 | 85822.2 | 1 | 436 | 124 | 5.43E-05 | 85822.2 | 1 | 12 | 36 | 3.49E-03 | 85822.2 |
| 14 | CHNROSNB† | 50 | 4 | 15049 | 255051 | 1.54E+02 | 592.4 | 2 | 3490 | 726 | 2.21E-04 | 4.16E-010 | 1 | 226 | 5050 | 2.14E-11 | 6.51E-22 |
| 15 | CLIFF | 2 | 4 | 1690 | 603 | 9.80E-04 | 0.2 | 1 | 211 | 168 | 4.45E-10 | 0.2 | 1 | 30 | 82 | 3.98E-09 | 0.2 |
| 16 | CUBE† | 2 | 4 | 707 | 603 | 3.89E+00 | 4.61 | 1 | 166 | 33 | 5.69E-10 | 4.02E-021 | 1 | 51 | 103 | 2.16E-08 | 3.22E-19 |
| 17 | DECONVU† | 61 | 2 | 88 | 744 | 1.67E+01 | 14.84 | 2 | 11987 | 1732 | 4.05E-05 | 5.56E-008 | 1 | 2040 | 122637 | 9.98E-07 | 6.24E-09 |
| 18 | DENSCHNA†‡ | 2 | 1 | 100 | 66 | 7.74E-07 | 6.40E-13 | 1 | 31 | 19 | 4.85E-07 | 2.88E-013 | 1 | 9 | 19 | 6.78E-12 | 1.15E-23 |
| 19 | DENSCHNB†‡ | 2 | 1 | 240 | 147 | 1.15E-07 | 3.51E-15 | 1 | 31 | 14 | 3.80E-08 | 3.61E-016 | 1 | 15 | 16 | 2.83E-07 | 1.31E-14 |
| 20 | DENSCHNC† | 2 | 1 | 237 | 141 | 7.28E-07 | 1.02E-13 | 1 | 71 | 18 | 4.06E-08 | 0.18 | 1 | 13 | 31 | 7.75E-10 | 2.24E-20 |
| 21 | DENSCHND† | 3 | 2 | 32 | 20 | 5.98E+05 | 815702.87 | 1 | 217 | 86 | 3.05E-07 | 1.52E-011 | 1 | 51 | 181 | 6.46E-07 | 1.55E-10 |
| 22 | DENSCHNE | 3 | 2 | 12 | 20 | 2.99E+00 | 3.29 | 4 | 1506 | 301 | 9.24E-02 | 0 | 1 | 36 | 57 | 1.68E-10 | 7.05E-21 |
| 23 | DENSCHNF† | 2 | 2 | 201 | 111 | 2.03E-06 | 1.07E-14 | 1 | 47 | 16 | 1.07E-08 | 5.91E-019 | 1 | 9 | 19 | 6.43E-10 | 6.82E-22 |
| 24 | DIXMAANK† | 15 | 2 | 19 | 48 | 3.24E+01 | 138.96 | 1 | 965 | 944 | 2.98E-07 | 1 | 1 | 28 | 193 | 7.64E-12 | 39845 |
| 25 | DJTL | 2 | 4 | 779 | 603 | 8.45E+02 | -4991.28 | 4 | 2666 | 201 | 1.77E+07 | -6408.96 | 2 | 384 | 349 | 2.39E+06 | -5283.92 |
| 26 | ENGVAL2† | 3 | 4 | 1064 | 1204 | 7.46E+00 | 2.53 | 3 | 306 | 63 | 1.41E-06 | 2.96E-016 | 1 | 18 | 53 | 9.39E-07 | 1.93E-16 |
| 27 | ERRINROS† | 50 | 2 | 62 | 255 | 1.28E+04 | 19177.39 | 2 | 2570 | 451 | 8.67E-04 | 39.9 | 1 | 108 | 1582 | 8.83E-10 | 40.4 |
| 28 | EXPFIT†‡ | 2 | 1 | 106 | 51 | 7.55E-08 | 0.24 | 1 | 74 | 25 | 2.65E-08 | 0.24 | 1 | 22 | 28 | 1.27E-11 | 0.24 |
| 29 | GROWTHLS | 3 | 1 | 9 | 16 | 7.11E-42 | 3542.15 | 1 | 12 | 2 | 0.00E+00 | 3542.15 | 4 | 304 | 1201 | 1.75E+02 | 39.9 |
| 30 | GULF† | 3 | 4 | 1519 | 1204 | 9.15E-02 | 5.78 | 1 | 585 | 130 | 4.32E-09 | 4.09E-019 | 4 | 305 | 1201 | 3.28E-01 | 2.21 |
| 31 | HAIRY† | 2 | 1 | 475 | 252 | 1.41E-05 | 39864 | 4 | 596 | 201 | 1.02E-01 | 20 | 1 | 64 | 133 | 1.84E-16 | 39864 |
| 32 | HATFLDD | 3 | 1 | 16 | 20 | 0.00E+00 | 14.04 | 1 | 388 | 72 | 1.56E-09 | 2.55E-007 | 1 | 32 | 93 | 7.76E-09 | 6.62E-08 |
| 33 | HATFLDE | 3 | 2 | 162 | 116 | 1.60E-03 | 1.54E-004 | 3 | 134 | 41 | 4.99E-06 | 1.20E-04 | 1 | 38 | 101 | 5.09E-07 | 5.12E-07 |
| 34 | HEART6LS | 6 | 2 | 30 | 63 | 6.79E+01 | 101.9 | 4 | 3077 | 601 | 8.29E+04 | 16.72 | 4 | 607 | 4201 | 7.82E+00 | 9.78 |
| 35 | HEART8LS | 8 | 2 | 749 | 2682 | 2.04E+01 | 28.61 | 4 | 5687 | 801 | 1.30E-01 | 1.14 | 1 | 185 | 1486 | 4.06E-12 | 7.09E-25 |
| 36 | HELIX | 3 | 4 | 1511 | 1204 | 5.76E-01 | 0.14 | 4 | 311 | 301 | 5.42E+00 | 6.05 | 1 | 26 | 57 | 9.90E-10 | 3.24E-21 |
| 37 | HIELOW | 3 | 5 | 25 | 32 | nan | NAN | 5 | 5 | 2 | nan | NAN | 1 | 14 | 29 | 5.13E-04 | 874.17 |
| 38 | HILBERTA† | 2 | 1 | 4 | 4 | 1.31E-08 | 1.55E-16 | 1 | 168 | 166 | 9.55E-07 | 8.92E-012 | 1 | 4 | 4 | 1.31E-08 | 1.55E-16 |
| 39 | HILBERTB† | 10 | 1 | 12 | 12 | 3.42E-07 | 2.50E-14 | 1 | 33 | 12 | 3.14E-07 | 8.58E-015 | 1 | 12 | 12 | 3.42E-07 | 2.50E-14 |
| 40 | HIMMELBB | 2 | 3 | 42 | 33 | 1.17E-05 | 3.91E-15 | 1 | 20 | 14 | 4.46E-07 | 1.78E-013 | 1 | 23 | 34 | 1.04E-13 | 6.17E-33 |
| 41 | HIMMELBF | 4 | 4 | 1398 | 2005 | 7.59E+01 | 5684.33 | 2 | 168 | 50 | 1.13E-02 | 319.72 | 1 | 14 | 46 | 3.36E-05 | 318.57 |

Continued in Table 4...

Table 4: The results obtained with UNCMIN and TENMIN (continued).

| | | | TENMIN | | | | | UNCMIN-BFGS | | | | | UNCMIN-FD | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *no* | *prob* | *n* | *E* | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | *F* | *E* | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | *F* | *E* | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | *F* |
| 42 | HIMMELBG$^\dagger$ | 2 | 1 | 41 | 27 | 0.00E+00 | 0 | 1 | 20 | 10 | 7.27E-07 | 7.42E-014 | 1 | 14 | 16 | 6.48E-12 | 8.01E-24 |
| 43 | HIMMELBH$^{\dagger\ddagger}$ | 2 | 1 | 188 | 102 | 4.01E-07 | -1 | 1 | 29 | 13 | 9.85E-11 | -1 | 1 | 9 | 7 | 1.21E-09 | -1 |
| 44 | HUMPS$^\dagger$ | 2 | 4 | 1202 | 603 | 6.08E-02 | 0.01 | 4 | 449 | 201 | 1.24E-05 | 7.66E-010 | 4 | 247 | 601 | 3.52E+01 | 2152.31 |
| 45 | HYDC20LS | 99 | 2 | 103 | 300 | 2.72E+08 | 63399205.8 | 2 | 2690 | 514 | 2.61E+03 | 1118.48 | 4 | 10000 | 990001 | 2.66E+03 | 1094.9 |
| 46 | JENSMP | 2 | 1 | 99 | 60 | 1.37E-06 | 124.36 | 1 | 6 | 2 | 8.52E-33 | 2020 | 1 | 12 | 28 | 3.88E-06 | 124.36 |
| 47 | KOWOSB$^{\dagger\ddagger}$ | 4 | 4 | 1309 | 2005 | 4.30E-04 | 4.12E-004 | 1 | 272 | 72 | 1.70E-07 | 3.08E-04 | 1 | 16 | 41 | 1.76E-08 | 0 |
| 48 | LOGHAIRY$^{\dagger\ddagger}$ | 2 | 4 | 662 | 603 | 1.16E-01 | 0.18 | 1 | 281 | 129 | 1.63E-07 | 0.18 | 1 | 311 | 544 | 2.27E-13 | 0.18 |
| 49 | MANCINO$^\dagger$ | 100 | 2 | 109 | 404 | 7.66E+08 | 1.09E+12 | 2 | 614 | 250 | 2.36E-01 | 1.62E-007 | 2 | 117 | 1011 | 2.73E-08 | 5.54E-22 |
| 50 | MARATOSB$^\dagger$ | 2 | 4 | 1747 | 603 | 8.08E-02 | 1 | 2 | 29 | 6 | 9.03E-02 | 1 | 4 | 319 | 601 | 4.93E+01 | 0.78 |
| 51 | MEXHAT | 2 | 4 | 1127 | 603 | 5.10E+01 | 0.15 | 3 | 62 | 10 | 5.34E+01 | 0.38 | 1 | 33 | 88 | 1.37E-08 | -0.04 |
| 52 | MEYER3 | 3 | 4 | 2029 | 1204 | 3.28E+08 | 7422089.89 | 3 | 150 | 28 | 1.43E+02 | 112123.44 | 4 | 315 | 1201 | 1.42E+02 | 111980797 |
| 53 | OSBORNEA$^{\dagger\ddagger}$ | 5 | 2 | 1407 | 2556 | 2.83E+00 | 0.05 | 2 | 605 | 138 | 8.32E-05 | 7.70E-005 | 4 | 509 | 3001 | 3.56E-02 | 0.05 |
| 54 | OSBORNEB | 11 | 2 | 228 | 876 | 1.50E-01 | 0.21 | 5 | 572 | 158 | nan | NAN | 1 | 838 | 9889 | 7.90E-07 | 0.04 |
| 55 | OSCIPATH$^\dagger$ | 15 | 4 | 4501 | 24016 | 1.05E+00 | 1.09 | 2 | 830 | 146 | 7.14E-05 | 0.98 | 1 | 29 | 129 | 1.69E-07 | 0.98 |
| 56 | PALMER1C | 8 | 2 | 12 | 27 | 4.64E+04 | 44938.69 | 2 | 661 | 201 | 1.71E+00 | 168.83 | 1 | 15 | 55 | 4.30E-08 | 0.1 |
| 57 | PALMER1D | 7 | 2 | 11 | 24 | 5.41E+03 | 39348.14 | 2 | 2093 | 527 | 2.35E-01 | 27.94 | 1 | 11 | 25 | 4.35E-08 | 0.65 |
| 58 | PALMER2C | 8 | 2 | 12 | 27 | 4.26E+03 | 1841.64 | 2 | 602 | 172 | 2.54E+00 | 98.07 | 1 | 13 | 37 | 1.26E-09 | 0.01 |
| 59 | PALMER3C | 8 | 2 | 12 | 27 | 1.60E+03 | 349.95 | 2 | 675 | 155 | 1.69E+00 | 54.31 | 1 | 13 | 37 | 1.76E-09 | 0.02 |
| 60 | PALMER4C | 8 | 2 | 12 | 27 | 1.62E+03 | 325.07 | 3 | 781 | 160 | 1.85E+00 | 62.27 | 1 | 12 | 28 | 6.98E-08 | 0.05 |
| 61 | PALMER5C$^{\dagger\ddagger}$ | 6 | 2 | 10 | 21 | 1.57E-04 | 2.13 | 2 | 191 | 173 | 3.20E-06 | 2.13 | 1 | 9 | 15 | 4.43E-13 | 2.13 |
| 62 | PALMER6C | 8 | 2 | 12 | 27 | 2.94E+02 | 197.25 | 4 | 820 | 801 | 2.91E+01 | 198.87 | 4 | 809 | 7201 | 2.44E-04 | 0.04 |
| 63 | PALMER7C | 8 | 2 | 12 | 27 | 8.53E+02 | 293.16 | 3 | 730 | 165 | 3.40E+00 | 56.9 | 4 | 809 | 7201 | 2.33E-03 | 1.5 |
| 64 | PALMER8C | 8 | 2 | 12 | 27 | 3.67E+02 | 182.33 | 2 | 1283 | 225 | 3.34E-02 | 3.13 | 1 | 14 | 46 | 5.89E-10 | 0.16 |
| 65 | PENALTY2 | 200 | 1 | 202 | 202 | 5.95E+05 | 4.71E+13 | 1 | 201 | 1 | 2.01E+06 | 4.71E+013 | 1 | 201 | 1 | 2.01E+06 | 4.71E+13 |
| 66 | ROSENBR$^\dagger$ | 2 | 4 | 937 | 603 | 7.59E-01 | 0.16 | 1 | 247 | 57 | 5.73E-08 | 2.15E-016 | 1 | 35 | 73 | 1.30E-07 | 1.19E-17 |
| 67 | S308$^\dagger$ | 2 | 1 | 160 | 105 | 2.17E-07 | 0.77 | 1 | 44 | 21 | 4.06E-07 | 0.77 | 1 | 13 | 28 | 7.02E-12 | 0.77 |
| 68 | SENSORS | 100 | 2 | 110 | 404 | 4.21E+01 | -126.14 | 3 | 4044 | 700 | 1.89E-03 | -2088.28 | 1 | 152 | 3031 | 2.16E-04 | -1944 |
| 69 | SINEVAL$^\dagger$ | 2 | 4 | 1015 | 603 | 4.76E+00 | 3.12 | 4 | 381 | 201 | 3.25E-06 | 2.59E-014 | 1 | 79 | 154 | 5.52E-16 | 9.22E-33 |
| 70 | SISSER$^{\dagger\ddagger}$ | 2 | 1 | 89 | 87 | 3.20E-09 | 5.29E-13 | 4 | 204 | 201 | 3.27E-05 | 1.75E-007 | 1 | 17 | 43 | 4.84E-07 | 4.16E-10 |
| 71 | SNAIL$^{\dagger\ddagger}$ | 2 | 1 | 166 | 120 | 1.53E-09 | 1.14E-18 | 1 | 15 | 9 | 3.65E-13 | 3.32E-026 | 1 | 143 | 262 | 2.54E-07 | 2.54E-14 |
| 72 | STRATEC | 10 | 2 | 25 | 66 | 2.57E+02 | 2639.42 | 5 | 26 | 7 | nan | NAN | 1 | 35 | 221 | 4.79E-08 | 2212.26 |
| 73 | TOINTGOR$^\dagger$ | 50 | 2 | 54 | 153 | 2.38E+01 | 1578.08 | 2 | 5656 | 1191 | 9.25E-04 | 1373.91 | 1 | 57 | 307 | 2.19E-06 | 1373.91 |
| 74 | TOINTPSP$^\dagger$ | 50 | 2 | 3338 | 64311 | 2.00E+01 | 999.63 | 2 | 2946 | 529 | 1.34E-04 | 225.56 | 1 | 69 | 613 | 3.60E-06 | 225.56 |
| 75 | TOINTQOR$^\dagger$ | 50 | 1 | 52 | 52 | 2.18E-06 | 1175.47 | 1 | 1058 | 207 | 2.96E-04 | 1175.47 | 1 | 52 | 52 | 2.18E-06 | 1175.47 |
| 76 | VARDIM | 200 | 2 | 204 | 603 | 8.73E+14 | 1.12E+16 | 4 | 20205 | 20001 | 8.49E+08 | 108228497 | 1 | 230 | 5830 | 3.68E-11 | 5.38E-24 |
| 77 | VAREIGVL$^\dagger$ | 50 | 2 | 54 | 153 | 6.13E+00 | 51.95 | 1 | 171 | 61 | 8.00E-07 | 4.41E-013 | 1 | 90 | 919 | 3.17E-13 | 6.76E-28 |
| 78 | VIBRBEAM | 8 | 2 | 29 | 63 | 2.08E+07 | 6939.6 | 3 | 79 | 8 | 2.74E+06 | 1262.99 | 4 | 810 | 7201 | 3.02E+01 | 3.84 |
| 79 | WATSON$^\dagger$ | 12 | 2 | 22 | 65 | 3.08E+00 | 0.42 | 2 | 2158 | 421 | 6.31E-05 | 1.35E-005 | 4 | 1213 | 15601 | 9.72E-03 | 0 |
| 80 | YFITU$^\dagger$ | 3 | 4 | 1190 | 1204 | 3.57E+01 | 19.54 | 2 | 716 | 138 | 8.19E-06 | 6.67E-013 | 4 | 313 | 1201 | 5.93E-07 | 1.86E-09 |
| 81 | ZANGWIL2$^\dagger$ | 2 | 1 | 4 | 4 | 1.95E-09 | -18.2 | 1 | 6 | 3 | 0.00E+00 | -18.2 | 1 | 4 | 4 | 1.95E-09 | -18.2 |

$^\dagger$,$^\ddagger$: Problems included in the benchmark

Table 5: The results obtained with SR1-NC.

| | | | SR1-NC | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $no$ | $prob$ | $n$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | $F$ | $Err_H^{(7)}$ | $PD^{(8)}$ | $N_{ED}^{(9)}$ | $N_d^{(10)}$ | $r_{decd}^{(11)}$ | $r_{dt}^{(12)}$ | $N_{fb}^{(13)}$ |
| 1 | AKIVA‡ | 2 | 2 | 2 | 2 | nan | nan | nan | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | ALLINITU | 4 | 1 | 16 | 10 | 5.48E-07 | 5.74 | 2.66E-01 | 1 | 1 | 0 | 0 | 0 | 6 |
| 3 | ARGLINA | 200 | 1 | 3 | 2 | 3.07E-15 | 200 | 4.99E-01 | 1 | 1 | 0 | 0 | 0.4 | 1 |
| 4 | ARGLINB‡ | 200 | 2 | 53 | 7 | 6.78E+01 | 99.63 | 1.23E-13 | 0 | 0 | 0 | 0 | 0 | 46 |
| 5 | ARGLINC‡ | 200 | 2 | 52 | 6 | 8.43E+02 | 101.13 | 1.68E-13 | 0 | 0 | 0 | 0 | 0 | 46 |
| 6 | BARD | 3 | 1 | 35 | 22 | 8.05E-08 | 0.01 | 9.37E-03 | 1 | 3 | 2 | 0.06 | 0 | 13 |
| 7 | BEALE | 2 | 1 | 25 | 15 | 1.43E-09 | 2.38E-018 | 3.58E-05 | 1 | 2 | 1 | 0 | 0 | 10 |
| 8 | BIGGS6 | 6 | 1 | 58 | 31 | 4.74E-07 | 0.01 | 1.29E-01 | 0 | 7 | 6 | 0.16 | 0 | 27 |
| 9 | BOX3 | 3 | 1 | 15 | 12 | 6.43E-07 | 7.61E-014 | 8.30E-04 | 1 | 1 | 0 | 0 | 0 | 3 |
| 10 | BRKMCC | 2 | 1 | 11 | 6 | 3.37E-07 | 0.17 | 3.41E-04 | 1 | 1 | 0 | 0 | 0 | 5 |
| 11 | BROWNAL | 200 | 1 | 33 | 12 | 3.45E-08 | 9.09E-015 | 3.07E-04 | 1 | 1 | 0 | 0 | 0.11 | 21 |
| 12 | BROWNBS‡ | 2 | 2 | 15 | 10 | 1.31E+04 | 4.59E-005 | 2.77E-05 | 1 | 0 | 0 | 0 | 0 | 5 |
| 13 | BROWNDEN‡ | 4 | 2 | 74 | 16 | 3.25E-04 | 85822.2 | 3.79E-03 | 1 | 1 | 1 | 0 | 0 | 58 |
| 14 | CHNROSNB | 50 | 2 | 384 | 138 | 9.60E-07 | 4.06E-015 | 1.24E-01 | 1 | 36 | 36 | 0.01 | 0.08 | 246 |
| 15 | CLIFF‡ | 2 | 3 | 33 | 4 | 1.00E+00 | 35.15 | 2.68E+12 | 1 | 1 | 0 | 0 | 0 | 29 |
| 16 | CUBE | 2 | 1 | 84 | 34 | 9.50E-08 | 2.62E-018 | 2.08E-06 | 1 | 7 | 6 | 3.57E-005 | 0 | 50 |
| 17 | DECONVU | 61 | 1 | 167 | 65 | 3.01E-07 | 2.85E-009 | 4.72E-01 | 0 | 15 | 14 | 1.49E-005 | 0 | 102 |
| 18 | DENSCHNA | 2 | 1 | 15 | 9 | 1.01E-09 | 2.32E-019 | 2.04E-03 | 1 | 2 | 1 | 0 | 0 | 6 |
| 19 | DENSCHNB | 2 | 1 | 12 | 9 | 3.22E-09 | 1.60E-018 | 1.61E-05 | 1 | 1 | 0 | 0 | 0 | 3 |
| 20 | DENSCHNC | 2 | 1 | 27 | 12 | 5.96E-09 | 0.18 | 4.57E-02 | 1 | 2 | 1 | 1.58E-005 | 0 | 15 |
| 21 | DENSCHND | 3 | 1 | 129 | 66 | 4.26E-07 | 3.13E-010 | 5.13E+00 | 1 | 3 | 2 | 2.18E-012 | 0 | 63 |
| 22 | DENSCHNE | 3 | 1 | 34 | 16 | 2.90E-07 | 2.29E-014 | 1.37E-02 | 1 | 2 | 1 | 0.01 | 0 | 18 |
| 23 | DENSCHNF | 2 | 1 | 26 | 12 | 1.37E-07 | 7.95E-017 | 1.37E-04 | 1 | 1 | 0 | 0 | 0 | 14 |
| 24 | DIXMAANK | 15 | 1 | 91 | 52 | 1.33E-07 | 1 | 5.12E-03 | 1 | 7 | 6 | 0 | 0 | 39 |
| 25 | DJTL‡ | 2 | 2 | 423 | 128 | 1.81E-02 | -8951.54 | 1.25E-03 | 1 | 23 | 23 | 0.26 | 0 | 295 |
| 26 | ENGVAL2 | 3 | 1 | 71 | 34 | 2.33E-07 | 1.38E-016 | 1.42E-03 | 1 | 5 | 4 | 0.01 | 0 | 37 |
| 27 | ERRINROS | 50 | 2 | 609 | 190 | 4.26E-06 | 39.9 | 1.09E+00 | 1 | 30 | 30 | 0 | 0.14 | 419 |
| 28 | EXPFIT | 2 | 1 | 28 | 14 | 2.08E-07 | 0.24 | 1.10E-02 | 1 | 2 | 1 | 0 | 0 | 14 |
| 29 | GROWTHLS | 3 | 1 | 17 | 2 | 1.54E-89 | 3542.15 | 4.01E+92 | 0 | 1 | 0 | 0 | 0 | 15 |
| 30 | GULF | 3 | 1 | 48 | 25 | 8.62E-10 | 4.71E-020 | 1.79E-04 | 1 | 2 | 1 | 0.04 | 0 | 23 |
| 31 | HAIRY | 2 | 2 | 39 | 16 | 4.58E-06 | 20 | 2.55E-05 | 1 | 2 | 2 | 0.07 | 0 | 23 |
| 32 | HATFLDD | 3 | 1 | 41 | 20 | 1.31E-07 | 6.62E-008 | 1.01E-02 | 1 | 2 | 1 | 1.05E-008 | 0 | 21 |
| 33 | HATFLDE | 3 | 1 | 59 | 34 | 4.97E-09 | 2.73E-006 | 1.30E-02 | 1 | 4 | 3 | 0 | 0 | 25 |
| 34 | HEART6LS‡ | 6 | 2 | 1081 | 325 | 6.39E+00 | 0 | 2.79E-01 | 0 | 83 | 83 | 0.18 | 1 | 709 |
| 35 | HEART8LS‡ | 8 | 4 | 2327 | 801 | 7.41E-02 | 1.14 | 1.56E+01 | 0 | 185 | 185 | 0 | 1 | 1526 |
| 36 | HELIX | 3 | 1 | 76 | 36 | 4.77E-08 | 6.06E-018 | 3.61E-03 | 1 | 5 | 4 | 0 | 0 | 40 |
| 37 | HIELOW‡ | 3 | 2 | 2 | 2 | nan | nan | nan | 0 | 0 | 1 | 0 | 0 | 0 |
| 38 | HILBERTA | 2 | 1 | 4 | 4 | 1.43E-14 | 1.88E-027 | 0.00E+00 | 1 | 1 | 0 | 0 | 0 | 0 |
| 39 | HILBERTB | 10 | 1 | 10 | 6 | 2.06E-08 | 8.12E-017 | 6.85E-01 | 1 | 1 | 0 | 0 | 0 | 4 |
| 40 | HIMMELBB | 2 | 1 | 31 | 8 | 1.60E-08 | 3.59E-020 | 1.77E+01 | 0 | 1 | 0 | 0 | 0 | 23 |
| 41 | HIMMELBF | 4 | 2 | 47 | 21 | 3.92E-06 | 318.57 | 5.68E-03 | 1 | 1 | 1 | 2.18E-007 | 0 | 26 |

Continued in Table 6...

Table 6: The results obtained with SR1-NC (continued).

| | | | SR1-NC | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $no$ | $prob$ | $n$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|_\infty$ | $F$ | $Err_H$ | $PD$ | $N_{ED}$ | $N_d$ | $r_{decd}$ | $r_{dt}$ | $N_{fb}$ |
| 42 | HIMMELBG | 2 | 1 | 13 | 8 | 3.27E-07 | 2.00E-014 | 9.47E-04 | 1 | 2 | 1 | 0 | 0 | 5 |
| 43 | HIMMELBH | 2 | 1 | 9 | 7 | 5.64E-08 | -1 | 6.81E-05 | 1 | 1 | 0 | 0 | 0 | 2 |
| 44 | HUMPS | 2 | 1 | 261 | 73 | 3.44E-10 | 8.35E-019 | 1.85E-02 | 1 | 24 | 23 | 0.38 | 0 | 188 |
| 45 | HYDC20LS‡ | 99 | 2 | 558 | 21 | nan | nan | nan | 0 | 0 | 0 | 0 | 0 | 537 |
| 46 | JENSMP | 2 | 2 | 36 | 13 | 5.30E-06 | 259.58 | 1.33E-02 | 0 | 4 | 4 | 0.44 | 0 | 23 |
| 47 | KOWOSB | 4 | 1 | 32 | 24 | 2.01E-07 | 0 | 1.28E-02 | 1 | 2 | 1 | 0.01 | 0 | 8 |
| 48 | LOGHAIRY | 2 | 1 | 313 | 99 | 3.93E-10 | 0.18 | 1.62E-08 | 1 | 30 | 29 | 0.7 | 0 | 214 |
| 49 | MANCINO‡ | 100 | 2 | 777 | 82 | 6.68E+01 | 0 | 8.11E-01 | 0 | 16 | 16 | 2.74E-005 | 0.02 | 694 |
| 50 | MARATOSB‡ | 2 | 4 | 573 | 201 | 2.25E+01 | 0.94 | 2.68E-03 | 0 | 53 | 53 | 2.81E-007 | 0 | 372 |
| 51 | MEXHAT‡ | 2 | 2 | 71 | 14 | 2.67E+00 | -0.02 | 1.23E+02 | 1 | 0 | 0 | 0 | 0 | 57 |
| 52 | MEYER3‡ | 3 | 2 | 125 | 21 | 2.40E+07 | 62028.5 | 4.40E-02 | 0 | 4 | 4 | 2.89E-007 | 0 | 104 |
| 53 | OSBORNEA | 5 | 1 | 161 | 44 | 1.02E-09 | 5.46E-005 | 2.05E-03 | 1 | 9 | 8 | 1.00E-006 | 1 | 117 |
| 54 | OSBORNEB | 11 | 1 | 130 | 57 | 1.79E-08 | 0.04 | 3.56E-02 | 1 | 13 | 12 | 0.01 | 0 | 73 |
| 55 | OSCIPATH | 15 | 1 | 55 | 20 | 1.22E-07 | 0.98 | 1.79E-02 | 0 | 2 | 1 | 5.27E-006 | 0 | 35 |
| 56 | PALMER1C‡ | 8 | 3 | 56 | 8 | 1.68E+00 | 162.14 | 2.60E-09 | 1 | 1 | 0 | 0 | 1 | 48 |
| 57 | PALMER1D | 7 | 1 | 60 | 10 | 7.19E-10 | 0.65 | 4.68E-16 | 1 | 1 | 0 | 0 | 0 | 50 |
| 58 | PALMER2C | 8 | 1 | 49 | 12 | 6.10E-09 | 0.01 | 1.62E-16 | 1 | 1 | 0 | 0 | 0 | 37 |
| 59 | PALMER3C | 8 | 1 | 45 | 12 | 3.46E-09 | 0.02 | 5.19E-16 | 1 | 1 | 0 | 0 | 0 | 33 |
| 60 | PALMER4C | 8 | 1 | 45 | 12 | 3.11E-10 | 0.05 | 3.53E-16 | 1 | 1 | 0 | 0 | 0 | 33 |
| 61 | PALMER5C | 6 | 1 | 16 | 8 | 1.67E-11 | 2.13 | 2.05E-16 | 1 | 1 | 0 | 0 | 0 | 8 |
| 62 | PALMER6C‡ | 8 | 3 | 32 | 10 | 4.35E-04 | 0.1 | 3.54E-08 | 1 | 1 | 0 | 0 | 0 | 22 |
| 63 | PALMER7C | 8 | 1 | 39 | 12 | 9.17E-09 | 0.6 | 1.11E-16 | 1 | 1 | 0 | 0 | 0 | 27 |
| 64 | PALMER8C | 8 | 1 | 47 | 12 | 2.01E-09 | 0.16 | 2.89E-16 | 1 | 2 | 1 | 4.38E-007 | 0 | 35 |
| 65 | PENALTY2‡ | 200 | 2 | 3060 | 258 | 4.32E+00 | 3.61E+013 | 9.09E-02 | 1 | 32 | 32 | 0.48 | 0.15 | 2795 |
| 66 | ROSENBR | 2 | 1 | 80 | 39 | 8.19E-09 | 4.74E-020 | 8.03E-05 | 1 | 5 | 4 | 0.02 | 0 | 41 |
| 67 | S308 | 2 | 1 | 21 | 13 | 2.28E-08 | 0.77 | 3.72E-05 | 1 | 1 | 0 | 0 | 0 | 8 |
| 68 | SENSORS | 100 | 2 | 172 | 65 | 6.31E-07 | -2108.53 | 7.16E-01 | 1 | 11 | 11 | 0.44 | 0.04 | 107 |
| 69 | SINEVAL | 2 | 1 | 175 | 83 | 3.25E-09 | 3.75E-020 | 5.46E-06 | 1 | 20 | 19 | 0.35 | 0 | 92 |
| 70 | SISSER | 2 | 1 | 22 | 19 | 7.22E-07 | 8.54E-010 | 1.67E+03 | 1 | 1 | 0 | 0 | 0 | 3 |
| 71 | SNAIL | 2 | 1 | 17 | 12 | 5.41E-10 | 1.21E-019 | 9.30E-05 | 1 | 2 | 1 | 0 | 0 | 5 |
| 72 | STRATEC‡ | 10 | 2 | 37 | 4 | nan | nan | nan | 0 | 0 | 0 | 0 | 0 | 33 |
| 73 | TOINTGOR | 50 | 1 | 102 | 49 | 1.72E-07 | 1373.91 | 9.43E-02 | 1 | 7 | 6 | 0 | 0.25 | 53 |
| 74 | TOINTPSP | 50 | 1 | 141 | 62 | 8.51E-08 | 225.56 | 2.64E-02 | 1 | 11 | 10 | 0.04 | 0 | 79 |
| 75 | TOINTQOR | 50 | 1 | 44 | 31 | 1.78E-07 | 1175.47 | 1.59E-01 | 1 | 1 | 0 | 0 | 0.33 | 13 |
| 76 | VARDIM‡ | 200 | 5 | 66 | 15 | 6.68E+09 | 1.77E+009 | 2.85E-01 | 1 | 0 | 0 | 0 | 0 | 50 |
| 77 | VAREIGVL | 50 | 1 | 86 | 41 | 2.65E-07 | 3.13E-014 | 1.39E-01 | 0 | 4 | 3 | 0 | 0.33 | 45 |
| 78 | VIBRBEAM‡ | 8 | 2 | 247 | 42 | 1.36E-03 | 10.66 | 7.12E-02 | 1 | 10 | 10 | 0.03 | 1 | 205 |
| 79 | WATSON | 12 | 1 | 77 | 29 | 1.30E-07 | 9.34E-008 | 1.65E-03 | 0 | 4 | 3 | 2.80E-009 | 0 | 48 |
| 80 | YFITU | 3 | 2 | 208 | 78 | 9.89E-06 | 6.67E-013 | 1.86E-03 | 1 | 11 | 11 | 0.04 | 0 | 130 |
| 81 | ZANGWIL2 | 2 | 1 | 3 | 3 | 0.00E+00 | -18.2 | 5.36E-01 | 1 | 1 | 0 | 0 | 0 | 0 |

‡ $||\nabla f_*||_\infty > 1.0E\text{-}05$

Table 7: The results obtained with LSR1-NC.

| | | | L-BFGS | | | | | LSR1-NC | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $no$ | $prob$ | $n$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $PD$ | $N_{ED}$ | $N_d$ | $r_{decd}$ | $r_{dt}$ | $N_{fb}$ |
| 1 | AKIVA | 2 | 0 | 22 | 22 | 2.54E-06 | 6.17 | 1 | 10001 | 10001 | nan | nan | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | ALLINITU | 4 | 0 | 14 | 14 | 4.23E-07 | 5.74 | 0 | 21 | 13 | 1.23E-07 | 5.74 | 1 | 1 | 1 | 0.01 | 0 | 8 |
| 3 | ARGLINA | 200 | 0 | 4 | 4 | 2.81E-14 | 200 | 0 | 3 | 2 | 4.31E-14 | 200 | 1 | 0 | 0 | 0 | 0 | 1 |
| 4 | ARGLINB | 200 | -1 | 43 | 43 | 1.75E-01 | 99.63 | -1 | 116 | 3 | 1.31E+13 | 7.40E+011 | 0 | 0 | 0 | 0 | 0 | 112 |
| 5 | ARGLINC | 200 | -1 | 26 | 26 | 7.15E-04 | 101.13 | -1 | 116 | 3 | 2.30E+12 | 2.36E+010 | 0 | 0 | 0 | 0 | 0 | 112 |
| 6 | ARWHEAD | 5000 | 0 | 14 | 14 | 1.37E-04 | 1.22E-011 | 0 | 36 | 10 | 9.40E-05 | 1.11E-012 | 1 | 1 | 1 | 2.84E-006 | 0.25 | 26 |
| 7 | BARD | 3 | 0 | 23 | 23 | 1.23E-05 | 0.01 | 0 | 39 | 17 | 8.07E-06 | 0.01 | 1 | 5 | 5 | 0.06 | 0 | 22 |
| 8 | BDQRTIC | 5000 | -1 | 515 | 515 | 3.53E-03 | 20006.26 | -1 | 396 | 97 | 4.18E-03 | 20006.3 | 1 | 18 | 18 | 0.01 | 0.27 | 298 |
| 9 | BEALE | 2 | 0 | 15 | 15 | 7.97E-06 | 1.54E-011 | 0 | 27 | 17 | 2.94E-06 | 4.22E-013 | 1 | 0 | 0 | 0 | 0 | 10 |
| 10 | BIGGS6 | 6 | 0 | 46 | 46 | 3.71E-05 | 0.01 | 0 | 77 | 39 | 2.65E-05 | 0.01 | 0 | 6 | 6 | 0.07 | 0 | 38 |
| 11 | BOX3 | 3 | 0 | 12 | 12 | 3.75E-05 | 7.16E-007 | 0 | 12 | 9 | 2.42E-05 | 1.90E-007 | 1 | 0 | 0 | 0 | 0 | 3 |
| 12 | BRKMCC | 2 | 0 | 8 | 8 | 1.08E-08 | 0.17 | 0 | 11 | 6 | 3.34E-07 | 0.17 | 1 | 0 | 0 | 0 | 0 | 5 |
| 13 | BROWNAL | 200 | 0 | 13 | 13 | 2.29E-05 | 1.47E-009 | 0 | 84 | 15 | 3.93E-05 | 3.84E-011 | 1 | 2 | 2 | 1.28E-012 | 0 | 69 |
| 14 | BROWNBS | 2 | 0 | 25 | 25 | 9.13E+00 | 1.72E-010 | 0 | 19 | 14 | 9.36E-01 | 1.95E-012 | 1 | 0 | 0 | 0 | 0 | 5 |
| 15 | BROWNDEN | 4 | 0 | 27 | 27 | 5.82E-05 | 85822.2 | 0 | 80 | 19 | 4.81E-06 | 85822.2 | 1 | 1 | 1 | 9.02E-014 | 0 | 61 |
| 16 | BROYDN7D | 5000 | 0 | 1612 | 1612 | 3.85E-04 | 1987.63 | 1 | 28301 | 10001 | 5.52E+00 | 1136.93 | ? | 2578 | 2578 | 0.11 | 0.16 | 18300 |
| 17 | BRYBND | 1000 | 0 | 33 | 33 | 4.80E-06 | 1.08E-012 | 0 | 141 | 58 | 8.56E-06 | 3.25E-012 | 0 | 8 | 8 | 0 | 0.5 | 83 |
| 18 | CHAINWOO | 100 | 0 | 438 | 438 | 9.27E-05 | 1 | 0 | 2097 | 726 | 8.74E-05 | 4.57 | 1 | 203 | 203 | 0 | 0.38 | 1371 |
| 19 | CHNROSNB | 50 | 0 | 282 | 282 | 4.91E-05 | 4.12E-011 | 0 | 1378 | 506 | 7.02E-05 | 5.09E-011 | 1 | 112 | 112 | 0 | 0.13 | 872 |
| 20 | CLIFF | 2 | 0 | 41 | 41 | 1.22E-05 | 0.2 | 0 | 321 | 15 | 1.32E-05 | 0.2 | 1 | 2 | 1 | 1.76E-013 | 0 | 306 |
| 21 | COSINE | 10000 | 0 | 17 | 17 | 1.54E-03 | -9999 | 0 | 19 | 10 | 7.66E-04 | -9999 | 1 | 1 | 1 | 0.27 | 0 | 9 |
| 22 | CRAGGLVY | 5000 | 0 | 88 | 88 | 3.23E-04 | 1688.22 | 0 | 440 | 153 | 2.78E-04 | 1688.22 | 1 | 39 | 39 | 0 | 0.23 | 287 |
| 23 | CUBE | 2 | 0 | 50 | 50 | 3.32E-09 | 1.70E-019 | 0 | 110 | 45 | 2.16E-06 | 3.02E-015 | 1 | 4 | 4 | 2.76E-005 | 0 | 65 |
| 24 | CURLY10 | 10000 | 1 | 10001 | 10001 | 1.62E-01 | -1003162.71 | -1 | 9496 | 3309 | 1.21E+00 | -1.00E+006 | ? | 893 | 893 | 3.37E-005 | 0.45 | 6186 |
| 25 | DECONVU | 61 | 0 | 186 | 186 | 3.71E-05 | 1.40E-007 | 0 | 726 | 246 | 3.88E-05 | 3.71E-007 | 0 | 53 | 53 | 6.67E-007 | 0.6 | 480 |
| 26 | DENSCHNA | 2 | 0 | 11 | 11 | 3.56E-07 | 8.22E-014 | 0 | 16 | 11 | 1.34E-07 | 4.49E-015 | 1 | 0 | 0 | 0 | 0 | 5 |
| 27 | DENSCHNB | 2 | 0 | 9 | 9 | 6.39E-07 | 5.12E-014 | 0 | 11 | 8 | 3.47E-06 | 1.90E-012 | 1 | 0 | 0 | 0 | 0 | 3 |
| 28 | DENSCHNC | 2 | 0 | 17 | 17 | 2.83E-06 | 8.49E-013 | 0 | 32 | 14 | 1.34E-05 | 0.18 | 1 | 1 | 1 | 6.58E-006 | 0 | 18 |
| 29 | DENSCHND | 3 | 0 | 56 | 56 | 7.56E-06 | 1.18E-008 | 0 | 152 | 66 | 9.60E-06 | 3.78E-008 | 1 | 8 | 8 | 0 | 0 | 86 |
| 30 | DENSCHNE | 3 | 0 | 42 | 42 | 6.23E-06 | 9.72E-012 | 0 | 40 | 17 | 1.90E-07 | 9.03E-015 | 1 | 1 | 1 | 0.01 | 0 | 23 |
| 31 | DENSCHNF | 2 | 0 | 10 | 10 | 1.45E-07 | 6.81E-017 | 0 | 26 | 12 | 5.11E-08 | 5.53E-018 | 1 | 0 | 0 | 0 | 0 | 14 |
| 32 | DIXMAANA | 3000 | 0 | 14 | 14 | 2.70E-08 | 1 | 0 | 14 | 10 | 3.84E-07 | 1 | 1 | 0 | 0 | 0 | 0 | 4 |
| 33 | DIXMAANB | 3000 | 0 | 13 | 13 | 9.11E-07 | 1 | 0 | 12 | 8 | 8.10E-06 | 1 | 1 | 0 | 0 | 0 | 0 | 4 |
| 34 | DIXMAANC | 3000 | 0 | 14 | 14 | 5.55E-06 | 1 | 0 | 14 | 9 | 1.74E-06 | 1 | 1 | 0 | 0 | 0 | 0 | 5 |
| 35 | DIXMAAND | 3000 | 0 | 16 | 16 | 9.10E-06 | 1 | 0 | 18 | 10 | 9.11E-06 | 1 | 1 | 0 | 0 | 0 | 0 | 8 |
| 36 | DIXMAANE | 3000 | 0 | 262 | 262 | 8.75E-06 | 1 | 0 | 1290 | 517 | 8.12E-06 | 1 | 1 | 127 | 127 | 0 | 0.28 | 773 |
| 37 | DIXMAANF | 3000 | 0 | 239 | 239 | 9.24E-06 | 1 | 0 | 1250 | 458 | 6.63E-06 | 1 | 1 | 125 | 125 | 9.44E-005 | 0.33 | 792 |
| 38 | DIXMAANG | 3000 | 0 | 237 | 237 | 8.98E-06 | 1 | 0 | 2399 | 926 | 9.83E-06 | 1 | 1 | 244 | 244 | 0.01 | 0.34 | 1473 |
| 39 | DIXMAANH | 3000 | 0 | 231 | 231 | 8.91E-06 | 1 | 0 | 900 | 319 | 9.54E-06 | 1 | 1 | 85 | 85 | 7.13E-009 | 0.29 | 581 |
| 40 | DIXMAANI | 3000 | 0 | 1781 | 1781 | 2.09E-05 | 1 | 0 | 2198 | 824 | 9.86E-06 | 1 | 1 | 211 | 211 | 0.01 | 0.34 | 1374 |
| 41 | DIXMAANJ | 3000 | 0 | 302 | 302 | 1.48E-05 | 1 | 0 | 528 | 220 | 9.70E-05 | 1.12 | 1 | 51 | 51 | 0.18 | 0.27 | 308 |
| 42 | DIXMAANK | 15 | 0 | 56 | 56 | 7.48E-06 | 1 | 0 | 204 | 97 | 7.60E-06 | 1 | 1 | 14 | 14 | 0.33 | 0 | 107 |
| 43 | DIXMAANL | 3000 | 0 | 536 | 536 | 1.47E-05 | 1 | 0 | 1190 | 472 | 2.21E-05 | 1 | 1 | 104 | 104 | 0.12 | 0.36 | 718 |
| 44 | DIXON3DQ | 10 | 0 | 35 | 35 | 1.72E-05 | 1.23E-010 | 0 | 141 | 68 | 2.62E-05 | 1.93E-010 | 1 | 11 | 11 | 0.01 | 0 | 73 |
| 45 | DJTL | 2 | -1 | 159 | 159 | 2.71E+06 | -4804.81 | -1 | 3901 | 1250 | 2.50E-04 | -8951.54 | 1 | 159 | 159 | 0.12 | 0.14 | 2650 |
| 46 | DQDRTIC | 100 | 0 | 21 | 21 | 4.02E-06 | 4.07E-014 | 0 | 21 | 7 | 6.81E-09 | 1.27E-018 | 1 | 0 | 0 | 0 | 0 | 14 |
| 47 | DQRTIC | 5000 | 0 | 44 | 44 | 1.15E+00 | 3.01 | 0 | 244 | 80 | 1.94E+00 | 3.75 | 1 | 11 | 11 | 4.44E-006 | 0.16 | 164 |

Table 8: The results obtained with LSR1-NC (continued).

| | | | L-BFGS | | | | | LSR1-NC | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $no$ | $prob$ | $n$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $PD$ | $N_{ED}$ | $N_d$ | $r_{decd}$ | $r_{dt}$ | $N_{fb}$ |
| 48 | EDENSCH | 2000 | 0 | 32 | 32 | 3.50E-04 | 12003.28 | 0 | 142 | 48 | 1.14E-04 | 12003.3 | 1 | 6 | 6 | 0 | 0.38 | 94 |
| 49 | EG2 | 1000 | 0 | 5 | 5 | 1.46E-06 | -998.95 | 0 | 14 | 5 | 7.29E-07 | -998.95 | 1 | 0 | 0 | 0 | 0 | 9 |
| 50 | ENGVAL1 | 5000 | 0 | 18 | 18 | 3.40E-04 | 5548.67 | 0 | 55 | 30 | 2.02E-04 | 5548.67 | 1 | 3 | 3 | 9.85E-005 | 0.25 | 25 |
| 51 | ENGVAL2 | 3 | 0 | 36 | 36 | 1.65E-06 | 5.18E-016 | 0 | 87 | 43 | 2.22E-06 | 3.70E-016 | 1 | 4 | 4 | 0.1 | 0 | 44 |
| 52 | ERRINROS | 50 | 0 | 155 | 155 | 3.41E-04 | 39.9 | 0 | 587 | 214 | 3.38E-04 | 39.9 | 1 | 38 | 38 | 6.64E-005 | 0.67 | 373 |
| 53 | EXPFIT | 2 | 0 | 16 | 16 | 2.81E-06 | 0.24 | 0 | 37 | 17 | 3.51E-06 | 0.24 | 1 | 2 | 2 | 0.01 | 0 | 20 |
| 54 | EXTROSNB | 1000 | 0 | 4550 | 4550 | 3.37E-05 | 9.35E-009 | 0 | 142 | 56 | 2.00E-04 | 3.63E-011 | 0 | 9 | 9 | 5.94E-008 | 0.4 | 86 |
| 55 | FLETCBV2 | 5000 | 0 | 3 | 3 | 2.14E-04 | -0.5 | 0 | 1 | 1 | 4.41E-06 | -0.5 | 1 | 0 | 0 | 0 | 0 | 0 |
| 56 | FLETCBV3 | 5000 | 0 | 22 | 22 | 4.77E+02 | -4.12E+013 | 0 | 97 | 26 | 3.76E+01 | -3.47E+009 | 0 | 12 | 12 | 3.76E-005 | 0.24 | 71 |
| 57 | FLETCHBV | 5000 | 0 | 1341 | 1341 | 2.95E+10 | -7.57E+024 | -1 | 15244 | 5486 | 1.09E+10 | -1.70E+024 | ? | 1386 | 1386 | 0.2 | 0.31 | 9757 |
| 58 | FLETCHCR | 1000 | 0 | 5682 | 5682 | 1.26E-04 | 2.92E-011 | 0 | 189 | 67 | 2.86E-04 | 1.43E-010 | 1 | 12 | 12 | 0.05 | 0.33 | 122 |
| 59 | FMINSRF2 | 5625 | 0 | 287 | 287 | 2.59E-04 | 1 | 0 | 1828 | 677 | 2.49E-04 | 1 | 1 | 165 | 165 | 0.13 | 0.22 | 1151 |
| 60 | FMINSURF | 5625 | 0 | 645 | 645 | 9.95E-06 | 1 | 0 | 4113 | 1515 | 9.59E-06 | 1 | ? | 383 | 383 | 0.15 | 0.22 | 2598 |
| 61 | FREUROTH | 5000 | -1 | 45 | 45 | 6.59E-03 | 608159.19 | 0 | 56 | 23 | 4.55E-04 | 608159 | 1 | 1 | 1 | 8.20E-007 | 0.13 | 33 |
| 62 | GENROSE | 500 | 0 | 1229 | 1229 | 1.54E-04 | 1 | 0 | 7549 | 2671 | 2.10E-04 | 1 | 1 | 698 | 698 | 0.08 | 0.39 | 4878 |
| 63 | GROWTHLS | 3 | 0 | 204 | 204 | 2.64E-07 | 1 | 0 | 17 | 2 | 1.70E-89 | 3542.15 | 0 | 0 | 0 | 0 | 0 | 15 |
| 64 | GULF | 3 | 0 | 34 | 34 | 6.16E-04 | 0 | 0 | 97 | 45 | 4.71E-04 | 3.84E-005 | 1 | 5 | 5 | 0.05 | 0 | 52 |
| 65 | HAIRY | 2 | 0 | 145 | 145 | 1.80E-06 | 20 | 0 | 77 | 27 | 1.25E-10 | 20 | 1 | 4 | 4 | 0.19 | 0 | 50 |
| 66 | HATFLDD | 3 | 0 | 24 | 24 | 1.27E-07 | 6.62E-008 | 0 | 42 | 23 | 1.47E-05 | 6.62E-008 | 1 | 1 | 1 | 4.08E-008 | 0 | 19 |
| 67 | HATFLDE | 3 | 0 | 41 | 41 | 1.98E-05 | 5.12E-007 | 0 | 129 | 41 | 1.01E-05 | 5.12E-007 | 1 | 8 | 8 | 7.26E-007 | 0 | 88 |
| 68 | HEART6LS | 6 | -1 | 1557 | 1557 | 5.16E-01 | 16.65 | 0 | 6977 | 1613 | 4.37E-06 | 6.22E-016 | 1 | 402 | 337 | 0.03 | 0.17 | 5364 |
| 69 | HEART8LS | 8 | 0 | 1022 | 1022 | 2.79E-05 | 2.55E-010 | 0 | 10778 | 2943 | 4.18E-04 | 1.14 | 0 | 790 | 755 | 0.02 | 0.5 | 7835 |
| 70 | HELIX | 3 | 0 | 31 | 31 | 9.70E-06 | 1.72E-013 | 0 | 79 | 40 | 2.40E-06 | 4.04E-014 | 1 | 4 | 4 | 0 | 0 | 39 |
| 71 | HIELOW | 3 | -1 | 51 | 51 | 1.45E-05 | 874.17 | 1 | 10001 | 10001 | nan | nan | 0 | 0 | 0 | 0 | 0 | 0 |
| 72 | HILBERTA | 2 | 0 | 7 | 7 | 2.64E-09 | 2.76E-018 | 0 | 4 | 4 | 1.27E-13 | 1.24E-025 | 1 | 0 | 0 | 0 | 0 | 0 |
| 73 | HILBERTB | 10 | 0 | 7 | 7 | 6.29E-09 | 1.97E-018 | 0 | 9 | 5 | 2.69E-06 | 3.60E-013 | 1 | 0 | 0 | 0 | 0 | 4 |
| 74 | HIMMELBB | 2 | 0 | 20 | 20 | 6.06E-06 | 3.75E-010 | 0 | 31 | 8 | 7.81E-07 | 8.54E-017 | 0 | 0 | 0 | 0 | 0 | 23 |
| 75 | HIMMELBF | 4 | 0 | 26 | 26 | 1.16E-02 | 319.72 | 0 | 59 | 17 | 1.13E-02 | 319.7 | 1 | 4 | 4 | 0.03 | 0 | 42 |
| 76 | HIMMELBG | 2 | 0 | 13 | 13 | 1.15E-07 | 1.13E-015 | 0 | 8 | 8 | 6.75E-06 | 4.22E-012 | 1 | 0 | 0 | 0 | 0 | 0 |
| 77 | HIMMELBH | 2 | 0 | 6 | 6 | 9.60E-06 | -1 | 0 | 9 | 7 | 4.84E-08 | -1 | 1 | 0 | 0 | 0 | 0 | 2 |
| 78 | HUMPS | 2 | 0 | 239 | 239 | 3.54E-06 | 6.26E-011 | 0 | 714 | 223 | 8.15E-06 | 3.30E-010 | 1 | 88 | 88 | 0.38 | 0 | 491 |
| 79 | HYDC20LS | 99 | 1 | 10001 | 10001 | 1.75E+03 | 29.87 | 1 | 29135 | 10001 | 4.86E+02 | 61.79 | 0 | 2660 | 2660 | 0.03 | 0.26 | 19134 |
| 80 | INDEF | 5000 | -1 | 27 | 27 | 8.71E+01 | -3.19E+017 | 0 | 26 | 6 | 9.53E+01 | -2.11E+009 | 0 | 2 | 2 | 0.01 | 0.33 | 20 |
| 81 | JENSMP | 2 | 0 | 51 | 51 | 1.97E-07 | 124.36 | 0 | 137 | 34 | 1.44E-06 | 124.36 | 1 | 13 | 13 | 0.46 | 0 | 103 |
| 82 | KOWOSB | 4 | 0 | 46 | 46 | 3.21E-06 | 0 | 0 | 61 | 27 | 1.26E-06 | 0 | 1 | 5 | 5 | 0.02 | 0 | 34 |
| 83 | LIARWHD | 5000 | 0 | 26 | 26 | 4.33E-04 | 2.57E-012 | 0 | 60 | 22 | 1.10E-06 | 1.82E-016 | 1 | 3 | 3 | 0 | 0.22 | 38 |
| 84 | LOGHAIRY | 2 | 0 | 3 | 3 | 1.22E-03 | 6.55 | 0 | 1 | 1 | 1.74E-03 | 6.55 | 1 | 0 | 0 | 0 | 0 | 0 |
| 85 | MANCINO | 100 | 0 | 14 | 14 | 3.77E-04 | 1.81E-014 | 0 | 91 | 16 | 3.86E-04 | 1.88E-014 | 1 | 1 | 1 | 0 | 0 | 75 |
| 86 | MARATOSB | 2 | 0 | 1543 | 1543 | 5.65E-06 | -1 | -1 | 5807 | 2311 | 1.13E-05 | -1 | 1 | 467 | 467 | 9.52E-006 | 0.25 | 3495 |
| 87 | MEXHAT | 2 | 0 | 53 | 53 | 5.40E-06 | -0.04 | 0 | 428 | 156 | 1.10E-05 | -0.04 | 1 | 26 | 26 | 4.34E-007 | 0 | 272 |
| 88 | MEYER3 | 3 | -1 | 614 | 614 | 2.04E-01 | 87.95 | 0 | 8531 | 822 | 1.14E-03 | 87.95 | 1 | 202 | 57 | 1.88E-006 | 0 | 7709 |
| 89 | MOREBV | 5000 | 0 | 11 | 11 | 1.21E-04 | 5.44E-009 | 0 | 34 | 12 | 9.69E-05 | 4.23E-009 | 1 | 3 | 3 | 0 | 0.67 | 22 |
| 90 | MSQRTALS | 1024 | 0 | 2200 | 2200 | 2.23E-04 | 4.95E-007 | 0 | 12900 | 4732 | 1.72E-04 | 7.31E-006 | ? | 1237 | 1237 | 6.23E-005 | 0.11 | 8168 |
| 91 | MSQRTBLS | 1024 | 0 | 1702 | 1702 | 1.84E-04 | 1.71E-007 | 0 | 9354 | 3365 | 1.81E-04 | 1.07E-006 | ? | 846 | 846 | 0 | 0.14 | 5989 |
| 92 | NONCVXU2 | 5000 | 0 | 1858 | 1858 | 5.45E-02 | 11586.1 | 0 | 8087 | 2904 | 5.78E-02 | 11588.4 | ? | 775 | 775 | 6.45E-005 | 0.29 | 5183 |
| 93 | NONCVXUN | 5000 | 0 | 2713 | 2713 | 5.53E-02 | 11599.18 | 0 | 9768 | 3531 | 7.11E-02 | 11607.2 | ? | 917 | 917 | 7.12E-005 | 0.3 | 6237 |
| 94 | NONDIA | 5000 | 0 | 23 | 23 | 8.53E-07 | 4.59E-019 | 0 | 110 | 53 | 2.08E-05 | 3.52E-015 | 1 | 4 | 4 | 6.31E-005 | 0.15 | 57 |

Table 9: The results obtained with LSR1-NC (continued).

| | | | L-BFGS | | | | | LSR1-NC | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $no$ | $prob$ | $n$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $E$ | $N_f$ | $N_{\nabla f}$ | $\|\|\nabla f_*\|\|$ | $F$ | $PD$ | $N_{ED}$ | $N_d$ | $r_{decd}$ | $r_{dt}$ | $N_{fb}$ |
| 95 | NONDQUAR | 5000 | 0 | 184 | 184 | 6.40E-04 | 0 | 0 | 479 | 201 | 6.57E-04 | 0 | 1 | 38 | 38 | 0 | 0.38 | 278 |
| 96 | OSBORNEA | 5 | 0 | 142 | 142 | 1.57E-05 | 5.46E-005 | 0 | 431 | 118 | 1.80E-05 | 5.46E-005 | 1 | 25 | 20 | 0 | 0 | 313 |
| 97 | OSBORNEB | 11 | 0 | 200 | 200 | 8.91E-05 | 0.04 | 0 | 470 | 210 | 8.28E-05 | 0.04 | 1 | 41 | 41 | 0.03 | 0 | 260 |
| 98 | OSCIPATH | 15 | 0 | 12 | 12 | 9.94E-06 | 0.98 | 0 | 47 | 15 | 2.14E-05 | 0.98 | 0 | 1 | 1 | 1.33E-010 | 0 | 32 |
| 99 | PALMER1C | 8 | 1 | 10001 | 10001 | 5.26E+00 | 168.09 | 0 | 3222 | 771 | 2.51E-03 | 0.1 | 1 | 187 | 168 | 1.14E-007 | 0.14 | 2451 |
| 100 | PALMER1D | 7 | 1 | 10001 | 10001 | 1.45E+04 | 62.44 | 0 | 396 | 120 | 1.30E-03 | 0.65 | 1 | 28 | 27 | 0 | 0 | 276 |
| 101 | PALMER2C | 8 | 1 | 10001 | 10001 | 4.27E-01 | 4.4 | 0 | 1325 | 359 | 1.13E-03 | 0.01 | 1 | 88 | 81 | 1.43E-008 | 0.33 | 966 |
| 102 | PALMER3C | 8 | 1 | 10001 | 10001 | 3.14E-01 | 2.33 | 0 | 2465 | 684 | 8.67E-04 | 0.02 | 1 | 156 | 145 | 1.48E-009 | 0.4 | 1781 |
| 103 | PALMER4C | 8 | 1 | 10001 | 10001 | 9.56E+04 | 5536.66 | 0 | 1399 | 478 | 1.16E-03 | 0.05 | 1 | 79 | 73 | 3.88E-008 | 0.33 | 921 |
| 104 | PALMER5C | 6 | 0 | 14 | 14 | 2.78E-04 | 2.13 | 0 | 17 | 9 | 1.08E-05 | 2.13 | 1 | 0 | 0 | 0 | 0 | 8 |
| 105 | PALMER6C | 8 | 1 | 10001 | 10001 | 1.27E+01 | 0.1 | 0 | 2927 | 988 | 2.81E-03 | 0.02 | 1 | 205 | 195 | 1.06E-008 | 0.33 | 1939 |
| 106 | PALMER7C | 8 | 1 | 10001 | 10001 | 2.97E-01 | 5.4 | 0 | 1431 | 405 | 8.95E-03 | 0.7 | 1 | 88 | 83 | 7.02E-008 | 0.33 | 1026 |
| 107 | PALMER8C | 8 | 1 | 10001 | 10001 | 2.79E-01 | 3.01 | 0 | 1182 | 362 | 5.15E-04 | 0.16 | 1 | 70 | 62 | 2.68E-006 | 1 | 820 |
| 108 | PENALTY1 | 1000 | 0 | 79 | 79 | 5.19E-07 | 0.01 | 0 | 471 | 166 | 5.54E-06 | 0.01 | ? | 13 | 13 | 1.30E-021 | 0.11 | 305 |
| 109 | PENALTY2 | 200 | -1 | 106 | 106 | 1.65E+01 | 4.71E+013 | -1 | 222 | 48 | 1.26E+03 | 4.71E+013 | 1 | 8 | 8 | 0 | 0.67 | 173 |
| 110 | POWELLSG | 5000 | 0 | 54 | 54 | 1.64E-06 | 2.00E-011 | 0 | 147 | 56 | 8.97E-06 | 8.93E-010 | 1 | 5 | 5 | 0.02 | 0.28 | 91 |
| 111 | POWER | 10000 | 0 | 426 | 426 | 9.50E-06 | 1.67E-009 | 0 | 3148 | 1095 | 9.95E-06 | 6.07E-009 | ? | 269 | 269 | 1.75E-005 | 0.47 | 2053 |
| 112 | QUARTC | 5000 | 0 | 44 | 44 | 1.15E+00 | 3.01 | 0 | 244 | 80 | 1.94E+00 | 3.75 | 1 | 11 | 11 | 4.44E-006 | 0.25 | 164 |
| 113 | ROSENBR | 2 | 0 | 48 | 48 | 1.59E-07 | 2.78E-017 | 0 | 112 | 53 | 9.97E-06 | 7.51E-014 | 1 | 7 | 7 | 0.02 | 0 | 59 |
| 114 | S308 | 2 | 0 | 14 | 14 | 7.86E-07 | 0.77 | 0 | 24 | 16 | 1.85E-06 | 0.77 | 1 | 0 | 0 | 0 | 0 | 8 |
| 115 | SBRYBND | 5000 | 1 | 10001 | 10001 | 5.37E+05 | 24496.55 | 1 | 27716 | 10001 | 1.09E+06 | 40145.8 | ? | 2504 | 2504 | 0.13 | 0.35 | 17715 |
| 116 | SCHMVETT | 5000 | 0 | 35 | 35 | 4.94E-04 | -14994 | 0 | 185 | 70 | 4.83E-04 | -14994 | 1 | 13 | 13 | 2.14E-005 | 0.07 | 115 |
| 117 | SCOSINE | 5000 | -1 | 25 | 25 | 4.74E+10 | 2174.5 | 1 | 28270 | 10001 | 8.45E+07 | 1349.51 | ? | 2627 | 2627 | 0.08 | 0.37 | 18269 |
| 118 | SENSORS | 100 | 0 | 23 | 23 | 1.92E-05 | -2108.53 | 0 | 72 | 28 | 6.15E-04 | -2108.53 | 1 | 9 | 9 | 0.41 | 0 | 44 |
| 119 | SINEVAL | 2 | 0 | 97 | 97 | 1.99E-07 | 9.25E-018 | 0 | 286 | 128 | 7.11E-06 | 1.34E-014 | 1 | 21 | 21 | 0.17 | 0 | 158 |
| 120 | SINQUAD | 5000 | -1 | 60 | 60 | 1.11E-02 | -6757013.76 | -1 | 235 | 21 | 2.02E-02 | -6.76E+006 | 1 | 3 | 3 | 0.67 | 0.1 | 213 |
| 121 | SISSER | 2 | 0 | 12 | 12 | 5.89E-06 | 1.16E-008 | 0 | 26 | 23 | 8.71E-06 | 2.02E-008 | 1 | 0 | 0 | 0 | 0 | 3 |
| 122 | SNAIL | 2 | 0 | 143 | 143 | 4.21E-07 | 4.42E-014 | 0 | 19 | 12 | 2.15E-06 | 1.15E-012 | 0 | 1 | 1 | 7.46E-005 | 0 | 7 |
| 123 | SPARSINE | 5000 | 1 | 10001 | 10001 | 6.12E-04 | 5.38E-008 | 1 | 27281 | 10001 | 7.30E+00 | 0.71 | ? | 2507 | 2507 | 0.05 | 0.33 | 17280 |
| 124 | SPARSQUR | 10000 | 0 | 39 | 39 | 6.10E-06 | 1.57E-008 | 0 | 232 | 79 | 3.68E-06 | 5.49E-009 | 1 | 12 | 12 | 0 | 0.1 | 153 |
| 125 | SPMSRTLS | 4999 | 0 | 163 | 163 | 4.85E-04 | 2.84E-007 | 0 | 919 | 353 | 4.78E-04 | 3.90E-007 | 1 | 91 | 91 | 0.02 | 0.23 | 566 |
| 126 | SROSENBR | 5000 | 0 | 20 | 20 | 2.46E-06 | 4.76E-015 | 0 | 28 | 15 | 1.34E-04 | 9.74E-012 | 1 | 1 | 1 | 3.39E-006 | 0 | 13 |
| 127 | STRATEC | 10 | -1 | 65 | 65 | 1.19E+07 | -8474445.62 | 1 | 10034 | 10001 | nan | nan | 0 | 0 | 0 | 0 | 0 | 33 |
| 128 | TESTQUAD | 5000 | 0 | 5185 | 5185 | 9.44E-06 | 1.28E-012 | 1 | 28658 | 10001 | 3.91E-01 | 1.47E-005 | ? | 2712 | 2712 | 5.85E-005 | 0.53 | 18657 |
| 129 | TOINTGOR | 50 | 0 | 107 | 107 | 2.43E-04 | 1373.91 | 0 | 587 | 229 | 1.48E-04 | 1373.91 | 1 | 56 | 56 | 0.01 | 0.38 | 358 |
| 130 | TOINTGSS | 5000 | 0 | 18 | 18 | 1.90E-05 | 10 | 0 | 3 | 2 | 1.39E-40 | 10 | 1 | 0 | 0 | 0 | 0 | 1 |
| 131 | TOINTPSP | 50 | 0 | 102 | 102 | 2.76E-04 | 225.56 | 0 | 443 | 175 | 2.40E-04 | 225.56 | 1 | 40 | 40 | 0.29 | 0.33 | 268 |
| 132 | TOINTQOR | 50 | 0 | 37 | 37 | 1.30E-04 | 1175.47 | 0 | 127 | 50 | 1.11E-04 | 1175.47 | 1 | 6 | 6 | 2.16E-005 | 0 | 77 |
| 133 | TQUARTIC | 5000 | 0 | 26 | 26 | 1.60E-06 | 1.54E-014 | 0 | 36 | 17 | 9.02E-05 | 1.02E-013 | 1 | 1 | 1 | 0 | 0 | 19 |
| 134 | TRIDIA | 30 | 0 | 96 | 96 | 9.94E-06 | 1.80E-012 | 0 | 463 | 179 | 1.15E-05 | 1.72E-012 | 1 | 44 | 44 | 0 | 1 | 284 |
| 135 | VARDIM | 200 | 0 | 42 | 42 | 5.64E-06 | 2.96E-018 | 0 | 160 | 66 | 5.00E-07 | 2.33E-020 | 1 | 2 | 1 | 5.53E-013 | 1 | 94 |
| 136 | VAREIGVL | 50 | 0 | 25 | 25 | 2.56E-06 | 6.96E-013 | 0 | 114 | 48 | 9.60E-06 | 7.94E-012 | 1 | 7 | 7 | 2.43E-005 | 0 | 66 |
| 137 | VIBRBEAM | 8 | 1 | 10001 | 10001 | 1.83E-01 | 5.19 | -1 | 15103 | 4267 | 2.01E-01 | 10.36 | 1 | 1127 | 1091 | 0.01 | 0.13 | 10835 |
| 138 | WATSON | 12 | 0 | 627 | 627 | 9.74E-06 | 1.56E-007 | 0 | 760 | 310 | 8.38E-06 | 1.68E-007 | 0 | 71 | 71 | 0.04 | 0.33 | 450 |
| 139 | WOODS | 4000 | 0 | 115 | 115 | 5.40E-04 | 2.42E-009 | 0 | 76 | 28 | 2.78E-04 | 9.40E-011 | 1 | 3 | 3 | 0 | 0 | 48 |
| 140 | YFITU | 3 | 0 | 96 | 96 | 9.97E-04 | 1.62E-011 | 0 | 284 | 113 | 1.22E-04 | 6.97E-013 | 1 | 10 | 10 | 0 | 0 | 171 |
| 141 | ZANGWIL2 | 2 | 0 | 3 | 3 | 3.91E-15 | -18.2 | 0 | 3 | 3 | 0.00E+00 | -18.2 | 1 | 0 | 0 | 0 | 0 | 0 |

## References

[1] CUTEr web site. http://hsl.rl.ac.uk/cuter-www/.

[2] LAPACK web site. http://www.netlib.org/lapack/.

[3] SR1-NC source codes and the test output. http://students.sabanciuniv.edu/~figen/sr1nc/.

[4] TENMIN web site. http://www.netlib.org/toms/739.

[5] UNCMIN web site. http://people.sc.fsu.edu/~burkardt/f77_src/uncmin/uncmin.html.

[6] D.P. Bertsekas. *Nonlinear Programming (2nd Edition)*. Athena Scientific, Belmont, 1999.

[7] R. H. Byrd, H. F. Khalfan, and R. B. Schnabel. Analysis of a symmetric rank-one trust region method. *SIAM Journal on Optimization*, 6(4):1025–1039, 1996.

[8] A.R. Conn, N.I.M. Gould, and Ph.L. Toint. Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, 50:177–195, 1991.

[9] E. D. Dolan and J. J. More. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.

[10] M. C. Ferris, S. Lucidi, and M. Roma. Nonmonotone curvilinear line search methods for unconstrained optimization. *Computational Optimization and Applications*, 6(2):117–136, 1996.

[11] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1996.

[12] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. Technical Report TR/PA/01/04, CERFACS, 2004.

[13] N.I.M. Gould, S. Lucidi, M.Roma, and Ph.L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optimization Methods and Software*, 14:75–98, 2000.

[14] Jr. J. E. Dennis and J. J. Moreé. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19(1):46–89, 1977.

[15] P. T. Keenan. C++ and Fortran77 timing comparisons. Technical Report CRPC-TR93347, Rice University, 2004.

[16] C. T. Kelley. *Iterative Methods for Optimization*. Frontiers in Applied Mathematics, No:18. SIAM, 1999.

[17] C. T. Kelley and E. W. Sachs. Local convergence of the symmetric rank one iteration. *Computational Optimization and Applications*, 9:43–63, 1998.

[18] H. F. Khalfan, R. H. Byrd, and R. B. Schnabel. A theoretical and experimental study of the symmetric rank-one update. *SIAM Journal on Optimization*, 3(1):1–24, 1993.

[19] J. Nocedal. L-BFGS software. urlhttp://www.eecs.northwestern.edu/ nocedal/lbfgs.htm.

[20] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2006.

[21] A. Olivares, J. M. Moguerza, and F. J. Prieto. Nonconvex optimization using negative curvature within a modified linesearch. *European Journal of Operational Research*, 189:706–722, 2008.

[22] P.K.H. Phua. Eigenvalues and switching algorithms for quasi-newton updates. *Optimization*, 42:185–217, 1997.

[23] R. B. Schnabel, J. E. Koonatz, and B. E. Weiss. A modular system of algorithms for unconstrained minimization. *ACM Transactions on Mathematical Software*, 11:419–440, 1985.

[24] R. B. Schnabel, J. E. Koonatz, and B. E. Weiss. Algorithm 738: A software package for unconstrained optimization using tensor methods. *ACM Transactions on Mathematical Software*, 20:518–530, 1994.

[25] P. Spellucci. A modified rank one update which converges q-superlinearly. *Computational Optimization and Applications*, 19:273–296, 2001.

[26] T. L. Veldhuizen and M. Ed Jernigan. Will C++ be faster than Fortran? In Y. Ishikawa, R. R. Oldehoeft, J. V. W. Reynders, and M. Tholburn, editors, *Scientific Computing in Object-oriented Parallel Environments*. Springer, 1997.