# Comparison of ASCALCG versus CONMIN, CG_DESCENT, LBFGS and TN

**Neculai Andrei**
*Research Institute for Informatics,*
*Center for Advanced Modeling and Optimization,*
*8-10, Averescu Avenue, Bucharest 1, Romania.*
and
*Academy of Romanian Scientists,*
*54, Splaiul Independentei, Bucharest 5, Romania.*
*E-mail: nandrei@ici.ro*

Figue 1 presents the Dolan-Moré CPU time performance profiles of ASCALCG (accelerated BFGS preconditioned conjugate gradient algorithm by Andrei [1-5]) *versus* CONMIN (BFGS preconditioned conjugate gradient algorithm by Shanno and Phua [16-18]), CG_DESCENT (W) (a conjugate gradient algorithm with guaranteed descent with Wolfe line search by Hager and Zhang [7-11]), CG_DESCENT (AW) (a conjugate gradient algorithm with guaranteed descent with approximate Wolfe line search by Hager and Zhang [7-11]), LBFGS (m=3) (limited memory quasi-Newton algorithm by Nocedal [13] and Liu and Nocedal [12] and TN (truncated Newton algorithm by Schlick and Fogelson [14,15]) for a set of 750 unconstrained optimization test functions described in [6].

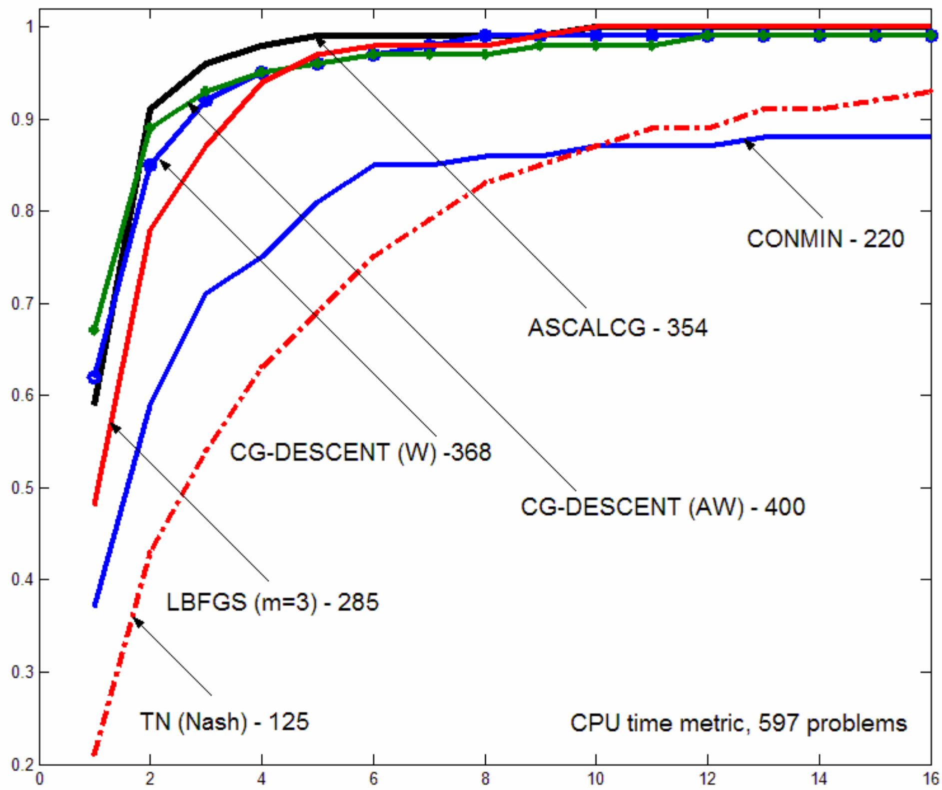


**Fig. 1.** Performance profiles of ASCALCG versus CONMIN, CG_DESCENT, LBFGS(m=3) and TN.

In Figure 1 for each algorithm we present the number of problems solved in minimum CPU time with dimensions ranging from 1000 to 10,000 variables. The left side of the figures gives the percentage of the test problems that were fastest solved by each of the algorithms. The top curve corresponds to the algorithm that solved the most problems in a time that was within a factor $\tau$ of the best time. Observe that for $\tau = 1$ the top curve in Figure 1 corresponds to CG_DESCENT (AW), this algorithm is clearly the fastest for this set of 597 test problems. The right side of the plots gives the percentage of the test problems that were successfully solved by each of the algorithms. Mainly, the right side is a measure of the robustness of an algorithm. In Figure 1 we see that ASCALCG and LBFGS (m=3) are the most robust algorithms we considered in this numerical study, ASCALCG being slight better.

## References


[1] Andrei, N., (2007a) *Scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* Optimization Methods and Software, 22 (2007), pp.561-571.

[2] Andrei, N., (2007b) *A scaled BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* Applied Mathematics Letters, 20 (2007), pp.645-650.

[3] Andrei, N., (2007c) *Scaled conjugate gradient algorithms for unconstrained optimization.* Computational Optimization and Applications, 38 (2007), pp.401-416.

[4] Andrei, N., (2007d) *A scaled nonlinear conjugate gradient algorithm for unconstrained optimization.* Optimization. A journal of mathematical programming and operations research, 57 (2008), pp.549-570.

[5] Andrei, N., (2008a) *Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization.* ICI Technical Report, March 24, 2008. [p16a08.pdf]

[6] Andrei, N., (2008b) *An unconstrained optimization test functions collection.* Advanced Modeling and Optimization. An Electronic International Journal, 10 (2008), pp.147-161.

[7] Hager, W.W., Zhang, H., (2003) *A new conjugate gradient method with guaranteed descent and an efficient line search*, University of Florida, Department of Mathematics, November 17, 2003 (theory and comparisons), revised July 3, 2004.

[8] Hager, W.W., Zhang, H., (2004) *CG-DESCENT, A conjugate gradient method with guaranteed descent (algorithm details and comparisons)*, University of Florida, Department of Mathematics, January 15, 2004.

[9] Hager, W.W., Zhang, H., (2005) *A new conjugate gradient method with guaranteed descent and an efficient line search*. SIAM Journal on Optimization, 16 (2005) 170-192.

[10] Hager, W.W., Zhang, H., (2006a) *Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent*. ACM Transactions on Mathematical Software, 32 (2006), 113-137.

[11] Hager, W.W., Zhang, H., (2006b) *A survey of nonlinear conjugate gradient methods,* Pacific Journal of Optimization, 2 (2006), pp.35-58.

[12] Liu, D.C., Nocedal, J., (1989) *On the limited memory BFGS method for large scale optimization.* Mathematical Programming, vol.45, 1989, pp.503-528.

[13] Nocedal, J., (1980) *Updating quasi-Newton matrices with limited starage.* Mathematics of Computation, 35 (1980), pp.773-782.

[14] Schlick, T, Fogelson, A., (1992a) *TNPACK - A truncated Newton minimization package for large-scale problems: I Algorithm and usage.* ACM Transactions on Mathematical Software, vol. 18, no.1, pp. 46-70, 1992.

[15] Schlick, T, Fogelson, A., (1992b) *TNPACK - A truncated Newton minimization package for large-scale problems: II Implementation examples.* ACM Transactions on Mathematical Software, vol. 18, no.1, pp. 71-111, 1992.

[16] Shanno, D.F., Phua, K.H., (1975) *Effective comparison of unconstrained optimization techniques.* Management Sciences, 22 (1975), pp.321-330.

[17] Shanno, D.F., Phua, K.H., (1976) *Algorithm 500, Minimization of unconstrained multivariate functions*, ACM Trans. on Math. Software, 2 (1976) 87-94.

[18] Shanno, D.F., Phua, K.H., (1980) *Remark on algorithm 500.* ACM Trans. on Math. Software, 6 (1980), pp. 618-622.


**July 22, 2008**