based on the Wolfe conditions (3.6) or strong Wolfe conditions (3.7), so that BFGS updating is stable.

The limited-memory BFGS algorithm can be stated formally as follows.

Algorithm 7.5 (L-BFGS).

Choose starting point x_0, integer $m > 0$;
$k \leftarrow 0$;
repeat
  Choose H_k^0 (for example, by using (7.20));
  Compute $p_k \leftarrow -H_k \nabla f_k$ from Algorithm 7.4;
  Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k is chosen to
    satisfy the Wolfe conditions;
  if $k > m$
    Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from storage;
  Compute and save $s_k \leftarrow x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$;
  $k \leftarrow k + 1$;
until convergence.

The strategy of keeping the m most recent correction pairs $\{s_i, y_i\}$ works well in practice; indeed no other strategy has yet proved to be consistently better. During its first $m - 1$ iterations, Algorithm 7.5 is equivalent to the BFGS algorithm of Chapter 6 if the initial matrix H_0 is the same in both methods, and if L-BFGS chooses $H_k^0 = H_0$ at each iteration.

Table 7.1 presents results illustrating the behavior of Algorithm 7.5 for various levels of memory m. It gives the number of function and gradient evaluations (nfg) and the total CPU time. The test problems are taken from the CUTE collection [35], the number of variables is indicated by n, and the termination criterion $\|\nabla f_k\| \leq 10^{-5}$ is used. The table shows that the algorithm tends to be less robust when m is small. As the amount of storage increases, the number of function evaluations tends to decrease; but since the cost of each iteration increases with the amount of storage, the best CPU time is often obtained for small values of m. Clearly, the optimal choice of m is problem dependent.

Because some rival algorithms are inefficient, Algorithm 7.5 is often the approach of choice for large problems in which the true Hessian is not sparse. In particular, a Newton

Table 7.1 Performance of Algorithm 7.5.

Problem	n	L-BFGS $m = 3$		L-BFGS $m = 5$		L-BFGS $m = 17$		L-BFGS $m = 29$	
		nfg	time	nfg	time	nfg	time	nfg	time
DIXMAANL	1500	146	16.5	134	17.4	120	28.2	125	44.4
EIGENALS	110	821	21.5	569	15.7	363	16.2	168	12.5
FREUROTH	1000	>999	—	>999	—	69	8.1	38	6.3
TRIDIA	1000	876	46.6	611	41.4	531	84.6	462	127.1