Chapter 6

Comparison of Direct Search Strategies for Parameter Optimization

6.1 Difficulties

The vast and steadily increasing number of optimization methods necessarily raises the question of which is the best strategy. There seems to be no unique answer. If indeed there were an optimal optimization method all the others would be superfluous and would have been long ago forgotten.

Because of the strong competition between already existing strategies it is necessary nowadays that whenever any proposal for a new method or variant is made, its advantages and improvements compared to older strategies be displayed. The usual way is to refer to a minimum problem for which the known methods fail to find a solution whereas the new proposal is successful. Or it is shown with reference to chosen examples that computation time or iterations can be saved by using the new version. The series of publications along these lines can in principle be continued indefinitely. With sufficient insight into the working of any strategy a special optimization problem can always be constructed for which the strategy fails. Likewise for any problem a special method of solution can be devised that is superior to the other procedures. One simply needs to exploit to the full what one knows of the problem structure as contained in its mathematical formulation.

Progress in the field of optimization methods does not, however, consist in developing an individual method of solution for each problem or type of problem. A practitioner would much rather manage with just one strategy, which can solve all the practically occurring problems for as small a total cost as possible. But as yet there is no such universal optimization method, and some authors doubt if there ever will be (Arrow and Hurwicz, 1957). All the methods presently known can only be used without restriction in particular areas of application. According to the nature of the particular problem, one or another strategy offers a more successful solution. The question of which is the best strategy is itself a kind of optimization problem. To be able to answer it objectively an objective function would have to be formulated for deciding which of two methods was best from the point of view of its results. So long as no generally recognized quality function of this kind exists, the question of which optimization method is optimal remains unanswered.

6.2 Theoretical Results

Classical optimization theory is concerned with establishing necessary and sufficient existence criteria for maxima and minima. It provides systems of equations but no iterative methods of finding their solutions. Not even Dantzig's simplex method (1966) for solving linear programming problems can be regarded as a direct result of theory–theoretical considerations of the linear problem only show that the extremum sought, except in special cases, must always lie in a corner of the polyhedron defined by the constraints. With n variables and m constraints (together with n non-negativity conditions) the number of corners or points of intersection of the hypersurfaces formed by the constraints is also limited to a maximum of $\binom{m+n}{n}$. Even the systematic inspection of all the points of intersection would be a finite optimization method. But not all the points of intersection are also within the allowed region (Saaty, 1955, 1963). Müller-Merbach (1971) gives $m\,n - m + 2$ as an upper bound to the number of feasible corner points. The simplex method, which is a method of steepest ascent along the edges of the polyhedron only traverses a tiny fraction of all the corners. Dantzig (1966) refers to empirical evidence that the number of necessary iterations increases as n, the number of variables, if the number of constraints m is constant, or as m if $(n - m)$ is not too small. Since, in the least favorable case, between m and $2\,m$ exchange operations must be performed on the tableau of $(m + 1)(n + 1)$ coefficients, the average computation time increases as $O(m^2\,n)$. In so-called degenerate cases, however, the simplex method can also become infinite. The repeated cycling through the same corners must then be broken by a rule for randomly choosing the iteration step (Dantzig). From a theoretical point of view the *ellipsoid method* of Khachiyan (1979) and the *interior point method* of Karmarkar (1984) do have the advantage of polynomial time consumption even in the worst case.

The question of finiteness of iterative methods is also a central theme of non-linear programming. In this case the solution can lie at any point on the border or interior of the enclosed region. For the special case that the objective function and all the constraint functions are convex and multiply differentiable Kuhn and Tucker (1951) and John (1948) have derived necessary and sufficient conditions for extremal solutions. Most of the iteration methods that have been developed on this basis are designed for problems with a quadratic objective function and linear constraints. Representative of quadratic programming are, for example, the methods of Beale (1956) and Wolfe (1959a). They make extensive use of the algorithm of the simplex method and thus belong, according to Hadley (1969), to the class of neighboring extremal point methods. Other strategies can move into the allowed region in the course of the iterations. As far as the constraints permit they take the direction of the gradient of the objective function. They are therefore known as *gradient methods of non-linear programming* (Kappler, 1967). As their name may suggest, however, they are not suitable for all non-linear problems. Their convergence can be proved at best for differentiable quasi-convex programs (Künzi, Krelle, and

Oettli, 1962). For these conditions the number of required iterations and rate of convergence cannot be stated in general. The same is true for the methods of Khachiyan (1979) and Karmarkar (1984). In the following chapters a short summary is attempted of the convergence properties of non-linear optimization methods in the unconstrained case (hill climbing methods).

6.2.1 Proofs of Convergence

A proof of convergence of an iterative method will aim to show that a sequence of iteration points $x^{(k)}$ tends monotonically with the index k towards the point x' which is sought:

$$\lim_{k\to\infty} \|x^{(k)} - x'\| \to 0$$

or

$$\|x^{(k)} - x'\| \le \varepsilon, \quad \varepsilon \ge 0, \quad \text{for } K(\varepsilon) \le k < \infty$$

If a finite accuracy of approximation is required, e.g., in terms of a distance from the solution measured by the Euclidean norm, the number of necessary iterations is usually finite.

In the case of optimization strategies it is shown that the first partial derivatives vanish at the point x':

$$\nabla F(x') = 0$$

This *necessary condition* for an extremum of a continuously differentiable function $F(x)$ is at the same time the termination criterion of the procedure. There are numerous convergence proofs of this kind covering a very wide range of minimization methods. A good survey is given by Polak (1971). It contains convergence proofs for, among others

- The Newton-Raphson method
  Assumption: $F(x)$ is twice continuously differentiable, $\nabla^2 F(x)$ has an inverse

- A generalized gradient method based on the method of steepest descent
  Assumption: $F(x)$ is once continuously differentiable

- A derivative-free method with local variation of the variables, similar to the Gauss-Seidel iteration method
  Assumption: $F(x)$ is continuously differentiable

In many optimization methods that deal with a function of several variables, each iteration consists of a number of one dimensional minimizations. For such a procedure to be finite it is not enough to show that the sequence of iteration points tends monotonically to the desired solution. The number of arithmetic operations in each iteration must also be finite. However, a line search may only become exact in the limit of infinitely many steps, while for the overall procedure to be finite, each one dimensional minimization must be terminated. This can result in the loss of convergence. Polak therefore distinguishes between *conceptual algorithms*, with an arbitrary number of calculation steps in one iteration, and *practical algorithms* in which this number is finite. To ensure the convergence of a practical method, one must usually introduce adaptive rules for the termination of subroutines that would in principle run forever (Nickel, 1967; Nickel and Ritter, 1972).

A further limitation to the predictive power of proofs of convergence arises from the properties of the point x' referred to above. Even if confusion of maxima and minima is eliminated, the approximate solution x' can still be a saddle point. To exclude this possibility, the second and sometimes even higher partial derivatives must be constructed and tested. It still always remains uncertain whether the solution that is finally found represents the global minimum or only a local minimum of the objective function. The only possibility of proving the global convergence of a sequential optimization method seems to be to require unimodality of the objective function. Then only one local optimum exists that is also a global optimum. Some global convergence properties are only possessed by a few simultaneous methods, such as for example the systematic grid method or the Monte-Carlo method. They place no continuity requirements on the objective function but the separation of the trial points must be significantly smaller than the distance between neighboring minima and the required accuracy. The fact that its cost rises exponentially with the number of variables usually precludes the practical application of such a method.

How does the convergence of the evolution strategy compare? For fixed step lengths, or more precisely for fixed variances $\sigma_i^2 > 0$ of the normally distributed mutation steps, there is always a positive probability of going from any starting point (e.g., a local minimum) to any other point with a better objective function value, provided that the separation of the points is finite. For the two membered method, Rechenberg (1973) gives necessary and sufficient conditions that the probability of success should exceed a specified value. Estimates of the computation cost can only be made for special objective functions. In this respect there are problems in determining the rules for controlling the mutation step lengths and deciding when the search is to be terminated. It is hard to reconcile the requirements for rapid convergence in one case and for a certain minimum probability of global convergence in another.

6.2.2 Rates of Convergence

While it may be of importance from a mathematical point of view to show that under certain assumptions a particular method leads with certainty to the objective, it is even more important to know how much computational effort is required, or what is the rate of convergence. The question of how fast an optimal solution is approached, or how many iterations are needed to reach a prescribed small distance from the objective, can only be answered for a few abstract methods and under even more restrictive assumptions. One distinguishes between first and second order convergence. Although some authors reserve the term quadratic convergence for the case when the solution of a quadratic problem is found within a finite number of iterations, it will be used here as a synonym for second order convergence. A sequence of iteration points $x^{(k)}$ *converges linearly* to x^* if it satisfies the condition

$$\|x^{(k)} - x^*\| \leq c\,\theta^k$$

where $0 \leq \theta < 1$ and $c < \infty$, constant. All methods which progress to the objective as a geometric progression in this way are said to display first order convergence. For a suitable choice of step lengths, e.g., following Polak (1971), the strategy of steepest descents satisfies this condition if the objective function is at least twice continuously differentiable and strictly convex in the neighborhood of the local minimum x^*.

A sequence $x^{(k)}$ is said to be *quadratically convergent* if it satisfies the condition

$$\|x^{(k+1)} - x^*\| \leq c' \, \|x^{(k)} - x^*\|^2 < 1$$

where $c' < \infty$, constant. Strategies providing iteration points such that the error after a step is proportional to the square of the preceding error exhibit second order convergence. The number of exact significant figures approximately doubles at each iteration.

If a Newton method converges, then it converges quadratically either if the objective function is four times continuously differentiable or if it is three times differentiable and the Hessian matrix of second partial derivatives is definite. Under the second condition it can be shown that the method of conjugate gradients with cyclic restart converges quadratically. If furthermore the objective function can be treated as convex, second order convergence can also be proved for the variable metric method. Polak (1971) shows that under the weaker assumption of a bounded Hessian matrix and an only twice continuously differentiable objective function, quadratic convergence can no longer be proved for the Newton-Raphson method. Its rate of convergence is still however greater than linear, i.e.,

$$\lim_{k \to \infty} \frac{\|x^{(k)} - x^*\|}{\theta^k} \to 0, \qquad \text{for } \theta \in (0, 1]$$

Quadratic convergence makes an optimization method attractive from a mathematical point of view. Unfortunately this desirable property is coupled with a tendency to diverge if the objective function is of higher than second order and the search is not started near the solution. For this reason combinations have often been proposed of a first order strategy at the start of an optimum search followed by a second order strategy in the neighborhood of the minimum.

6.2.3 Q-Properties

While it is to be expected that a quadratically convergent strategy will take fewer iterations to locate a minimum than one that only converges linearly, it is still of interest to know the explicit number of calculation steps required. This can only be given in a general form for the simplest case of a non-linear minimization problem, namely for quadratic objective functions

$$F(x) = x^T A \, x + b \, x + c$$

with a positive definite matrix of coefficients A. Since all second order methods also employ a quadratic function as an internal model of the objective function for the purpose of predicting suitable directions and sometimes also step lengths, they can at least in principle find the exact solution within a finite number of steps. They are referred to by their so-called *Q-properties*. Thus if a strategy takes p iteration steps for locating exactly the quadratic optimum it is said to have the property $Q\ p$.

The Newton-Raphson method, for example, takes only a single step because the second partial derivatives are constant over the whole $\mathbb{R}^n$ and all higher order derivatives vanish. If the iteration rule is followed exactly it gives the position of the minimum right at the first step without the necessity of a line search. As no objective function values need to be evaluated explicitly one also refers to it as an indirect optimization method. It has the property $Q\ 1$.

A conjugate gradients method, e.g., that of Fletcher and Reeves (1964), requires up to n cycles before a complete set of conjugate directions is assembled and a line search leads to the minimum. It therefore has the property $Q\ n$.

Powell's (1964) derivative-free search method of conjugate directions requires $n + 1$ line searches for determining each of the n direction vectors and thus has the property $Q\ n(n+1)$ or $Q\ O(n^2)$ in terms of the number of one dimensional minimizations.

The variable metric strategy of Davidon (1959) in the formulation of Fletcher and Powell (1963) can be interpreted both as a quasi-Newton method and as a method with conjugate directions. If the objective function is quadratic, then the iteratively improved approximate matrix agrees with the exact inverse of the Hessian matrix after n iterations. This method has the property $Q\ n$.

Apart from the fact that any practical algorithm can require more than the theoretically predicted number of iterations due to the effect of rounding errors, for peculiar types of coefficient matrix in the quadratic problem the algorithm can fail completely. For example Zangwill (1967) demonstrates such a source of error in the Powell method if no improvement is achieved in one direction.

6.2.4 Computing Demands

The specification of the Q-properties of individual strategies is only the first step towards estimating the computing demands. In different procedures an iteration or a cycle comprises various different operations. It is useful to distinguish ordinary calculation operations like additions and multiplications from the evaluation of functions such as the objective function and its derivatives. The number of variables is the basic quantity that determines the computation cost. A crude but adequate measure is therefore given by the power p of n, the number of parameters, with which the expected computation times increase. For the case of many variables, since the highest powers are dominant, lower order terms can be neglected. In the Newton-Raphson method, at each iteration the gradient vector ∇F and the Hessian matrix $\nabla^2 F$ must be evaluated, which means n first and $\frac{n}{2}(n + 1)$ second partial derivatives. Objective function values are not required. In fact the most costly step is the matrix inversion. It requires in the order of $O(n^3)$ operations. A cycle of the conjugate gradient method consists of a line search and a gradient determination. The one dimensional minimization requires several calls of the objective function. Their number depends on the choice of method but it can be regarded as constant, or at least as independent of the number of variables. The remaining steps in the calculation, including vector multiplications, are composed of $O(n)$ elementary arithmetical operations. Similar results apply in the case of the variable metric strategy, except that there are an additional $O(n^2)$ basic operations for matrix additions and multiplications. The direct search method due to Powell evaluates neither first nor second partial derivatives. After every $n + 1$ line searches the direction vectors are redefined, which requires $O(n^2)$ values to be assigned. But since each one dimensional optimization counts as an iteration step, only $O(n)$ direct operations are attributed to each iteration. A convenient summary of the relationships is given in Table 6.1. For simplicity only the terms of highest order in the number of parameters n are accounted for, without their coefficients of proportionality.

So far we have no scale for comparison of the different function evaluations with each other. Fletcher (1972a) and others consider an evaluation of the Hessian matrix to be equivalent to $O(n)$ gradient determinations or $O(n^2)$ objective function calls. This type of scaling is valid whenever the partial derivatives cannot be obtained in analytic form and provided as functions, but are calculated approximately as quotients of differences obtained by trial steps in the coordinate directions. In any case it ought to be about right if the objective function is of higher than second order. Accordingly the following weighting of the function evaluations can be introduced on the table:

$$F : \nabla F : \nabla^2 F \mathrel{\widehat{=}} n^0 : n^1 : n^2$$

Before anything can be said about the overall computation cost, or time, one must know how many operations are required for calculating a value of the objective function. In general a function of n variables will entail a cost that rises at least linearly with n.

Table 6.1: Number of operations required by the most important basic strategies to minimize a quadratic objective function in terms of the number of variables n (only orders of magnitude)

Strategy	Number of iterations	Number of operations per iteration			
		Function evaluations			Elementary operations
		F	∇F	$\nabla^2 F$	
Newton e.g., Newton-Raphson	n^0	—	n^0	n^0	n^3
Variable metric e.g., Davidon	n^1	n^0	n^0	—	n^2
Conjugate gradients e.g., Fletcher-Reeves	n^1	n^0	n^0	—	n^1
Conjugate directions e.g., Powell	n^2	n^0	—	—	n^1
		n^0	n^1	n^2	
		Weighting factors			

For a quadratic function with a full matrix of coefficients, just to evaluate the expression $x^T A\,x$ requires $O(n^2)$ basic arithmetical operations. If the order of magnitude is denoted by $O(n^f)$ then, assuming $f \geq 1$, for all the optimization methods considered so far the computation time is given by:

$$T \sim n^{2+f} \geq n^3$$

The advantage of having fewer function-independent operations in the Fletcher-Reeves method, therefore, only makes itself felt if the number of variables is small and the time for one function evaluation is short.

All the variants of the basic second order strategies mentioned here can be fitted, with similar assumptions, into the above scheme. Among these are (Broyden, 1972)

- Modified and quasi-Newton methods
- Methods of conjugate gradients and conjugate directions
- Variable metric strategies, with their variations using correction matrices of rank one

There is no optimization method that has a cost rising with less than the third power of the number of variables. Even the indirect procedure, in which the equations for the necessary conditions for an extremum are set up and solved by conventional methods, does not afford any basic reduction in the computational effort. If the objective function is quadratic, a system of n simultaneous linear equations is obtained. To solve for the n unknowns the Gaussian elimination method requires $\frac{1}{3}n^3$ basic operations (multiplications and divisions). According to Zurmühl (1965) all the other direct methods, meaning here non-iterative methods, are more costly, except in special cases. Methods involving a stepwise approach to the solution of systems of linear equations (relaxation methods) require an infinite number of iterations to reach an absolutely exact result. They converge linearly and correspond to first order optimization strategies (single step or Gauss-Seidel methods and total step or gradient methods; see Schwarz, Rutishauser, and Stiefel, 1968). Only the method of Hestenes and Stiefel (1952) converges after a finite number of calculation steps, assuming that the calculations are exact. It is a conjugate gradient method for solving systems of linear equations with a symmetrical, positive-definite matrix of coefficients.

The main concern here is with direct, i.e., derivative-free, search strategies for optimization. Finiteness of the search in the quadratic case and greater than linear convergence can only be proved for the Powell method of conjugate directions and for the Davidon-Fletcher-Powell variable metric method, which Stewart reformulated as a derivative-free quasi-Newton method. Of the coordinates strategy, at best it can be said that it converges linearly. The same holds for the simple gradient methods. There are also versions of them in which the partial derivatives are obtained numerically. Since various comparison tests have shown them to be rather ineffective in highly non-linear situations, none is considered here. No theoretically founded statements about convergence rates and Q-properties are available for the other direct strategies. The rate of progress defined by Rechenberg (1973) for the evolution strategy with adaptive step length control represents an average measure of convergence. It could, however, only be determined theoretically for two selected model objective functions. The one with concentric contour lines, or contour hypersurfaces, can be regarded as a special case of a quadratic objective function. The formula for the local rate of progress in both the two membered and the multimembered strategies has the form

$$\varphi(r) = c\,\frac{r}{n}\,, \qquad c = const.$$

r is the current distance from the objective:

$$r = \|x^{(k)} - x^*\|$$

and φ is the change in r at one iteration or mutation

$$\varphi(r) = \triangle r = \|x^{(k)} - x^*\| - \|x^{(k+1)} - x^*\|$$

Rearrangement of the above formulae gives

$$\|x^{(k+1)} - x^*\| = \|x^{(k)} - x^*\| \left(1 - \frac{c}{n}\right)$$

or

$$\|x^{(k)} - x^*\| = \|x^{(0)} - x^*\| \left(1 - \frac{c}{n}\right)^k$$

which because

$$0 < 1 - \frac{c}{n} < 1\,, \qquad \text{for } 1 \le n < \infty$$

proves the linear convergence property of the evolution strategy.

6.3 Numerical Comparison of Strategies

While the statements about convergence and rates of convergence derived from theory are not without value, they can say little about the capability of optimization methods in the general non-linear case because of the frequently rather limiting assumptions or restrictions. The computational effort for example could only be specified for quadratic objective functions. The need therefore arises for numerical tests even for mathematically based methods in the case of non-linear optimization. Many of the direct strategies are only heuristic in nature anyway. They owe their success simply to the experimental evidence of their usefulness in practical situations.

Iteration methods usually require a considerable number of calculation steps. Without mechanical assistance they frequently cannot be applied at all. There is thus an evident parallel between the development of rapid digital computers and optimization methods. The use of such systems entails, however, one difficulty. The possibly unpleasant consequences of finite accuracy in line searches have already been pointed out. The finite number of decimal places to which data are stored implies that all calculation operations are subject to rounding errors, unless they are dealing with integers. Proofs of convergence, however, assume that the calculations are performed exactly. They therefore only hold for the idealized concept of an algorithm, not for a particular computer program. The susceptibility of a strategy to rounding errors depends on how it is coded. Thus, for this reason too there is a need to check the convergence properties of numerical methods experimentally.

Because of the finite word length of a digital computer the number range is also limited. If it is exceeded, the program that is running normally terminates. Such fatal execution errors (floating overflow, floating divide check), are usually the consequence of rounding errors in previous steps; if the error is in going below the absolutely smallest number value (floating underflow) it is not regarded as fatal. Only few algorithms, e.g., Brent (1973), take special account of finite machine accuracy.

In spite of the frequent mention of the importance of numerical comparisons of strategies, few publications to date have reported results on several different test problems using a large number of minimization methods. By virtue of its scope, the work of Colville (1968, 1970) stands out among the older studies by Brooks (1959), Spang (1962), Dickinson (1964), Leon (1966a), Box (1966), and Kowalik and Osborne (1968). It included 30 strategies and 8 different problems, but not many direct search methods compared to gradient methods. In some other tests by Jacoby, Kowalik, and Pizzo (1972), Himmelblau (1972a), Smith (1973), and others in the collection of Lootsma (1972a), derivative-free strategies receive much more attention. The comparisons of Gorvits and Larichev (1971) and Larichev and Gorvits (1974) treat only gradient methods, and that of Tapley and Lewallen (1967) deals with some schemes for the numerical treatment of functional optimization problems. The huge collection of test problems of Hock and Schittkowski (1981) is biased towards standard methods of mathematical programming and their capabilities (Schittkowski, 1980).

6.3.1 Computer Used

The machine on which the numerical experiments were carried out was a PDP 10 from the firm Digital Equipment Corporation, Maynard, Massachusetts. It had the following specifications:

Core storage area:	64K (1K = 1024 words)
Word length:	36 bits
Cycle time:	1.65 or 1.8 μs

The time-sharing operating system accounted for about 34K of core, so that only 30K remained available to the user. To tackle some problems with as many variables as possible, the computations were generally worked only to single precision. The main program, which was the same for all strategies, occupied about $\left(5 + \frac{2n}{1024}\right)$ K words, and the FORTRAN library a further 5K. The consequent maximum number n_{max} of parameters is given for each search method under test in Table 6.2. The finite word length of a digital computer means that its number range is limited. The absolute bounds for floating point arithmetic were given by:

Largest absolute number:	$2^{127} \simeq 1.7 \cdot 10^{38}$
Smallest absolute number:	$2^{-128} \simeq 2.9 \cdot 10^{-39}$

Only a part of the word is available for the mantissa of a number. This imposed the differential accuracy limit, which is much lower and usually more important:

Smallest difference relative to unity: $2^{-27} \simeq 7.5 \cdot 10^{-9}$

Accordingly the following equalities hold for this computer:

$$\begin{aligned} \varepsilon &= 0, &\quad \text{for } |\varepsilon| < 2^{-128} \\ 1+\varepsilon &= 1, &\quad \text{for } |\varepsilon| < 2^{-27} \end{aligned}$$

These computer-specific data play a rôle when testing for zero or for the equality of two quantities. The same programs can therefore lead to different results on different computers.

Strategies are often judged by the computation time they require to achieve a result, for example, with a specified accuracy. The basic quantity for this purpose is the occupation time of the central processor unit (CPU). It also depends on the machine. Word lengths and cycle times are not enough to allow comparison between runs that were made on different computers. So-called MIX-times, which are average values of the duration of certain operations, also prove to be unsuitable, since the speed of calculation is so strongly dependent on the frequency of its individual steps. A method proposed by Colville (1968) has received wide recognition. Its design was particularly suited to optimization methods. According to this scheme, measured computation times are expressed relative to the time required for 10 consecutive inversions of a 40×40 matrix, using the FORTRAN program written by Colville. In our case this unit was around 110 seconds. Because of the time-sharing operation, with its rather variable load on the PDP 10, there were deviations of 10% and above on the reported CPU times. This was especially marked for short programs.

6.3.2 Optimization Methods Tested

One goal of this work is to compare evolution strategies with other derivative-free methods of continuous parameter optimization. To this end we consider not only direct search methods in the narrower sense, but also those methods that glean their required knowledge of partial derivatives by means of trial steps and finite difference methods. Altogether 14 strategies or versions of basic strategies are considered. Their names and abbreviations used for them are listed in Table 6.2. All tests were run on the PDP 10 mentioned in the previous section.

Finite computer accuracy implies that in the case of quadratic objective functions the iteration process could or should not be continued until the exact solution has been obtained. The decision when to terminate the optimum search is a necessary and often crucial component of any iterative method. Just as the procedures of the individual strategies differ, so too do their termination or convergence criteria. As a rule, the user is given the chance to exert an influence on the termination criterion by means of an input parameter defined as the required accuracy. It refers either to the values of the variables (change in x_i within one iteration or size of the step lengths s_i) or to values of the objective function. Both criteria harbor the danger that the search will be terminated prematurely, that is before arriving as close to the objective as is required. This is made clear by Figure 6.1.

Neither $\triangle x < \varepsilon_x$ nor $\triangle F < \varepsilon_F$ alone are sufficient conditions for being close to the solution x^*. The condition $\|\nabla F\| < \varepsilon_g$, which is often applied for gradient methods, can lead to termination of the search near a saddle point and is in any case not always appropriate in the presence of constraints or discontinuities. Thus the effectiveness of a convergence criterion is always closely linked to the procedure of a particular strategy and not automatically transferable to other strategies. Since each method converges to the optimum along a different path, in spite of having the same required accuracy, different methods do not finish the search with the same result. The termination criteria are also tested at different points in time and not always with the same frequency. These factors make it more difficult to compare the test results of different methods.

For this reason Himmelblau replaces the strategy specific termination criteria by the tests

$$\|x^{(k)} - x^*\| < \varepsilon_1$$

and

$$F(x^{(k)}) - F(x^*) < \varepsilon_2$$

after each iteration. He thereby obtains results that can be compared quite easily, but they are valid only for strategies deprived of one of their major components.

We have retained here the original termination criteria of all the search methods. The required accuracies were set as high as the computer permitted. The actual values used are given in Table 6.2. Their meaning can be found in the description of the strategies in Chapter 3.

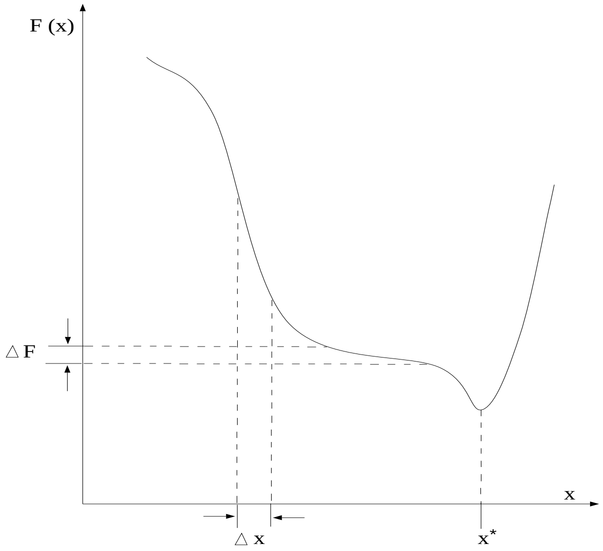

Figure 6.1: The adequacy of termination criteria

Table 6.2: Strategies applied: their abbreviations, maximum number of variables and accuracy parameters

Strategy	Abbreviation	Maximum number of variables	Accuracy parameter
Coordinate strategy with Fibonacci search	FIBO	2900	$\varepsilon = 7.5 \cdot 10^{-9}$
Coordinate strategy with golden section	GOLD	2910	$\varepsilon = 7.5 \cdot 10^{-9}$
Coordinate strategy with Lagrangian interpolation	LAGR	2540	$\varepsilon = 7.5 \cdot 10^{-9}$
Direct search of Hooke and Jeeves	HOJE	4090	$\varepsilon = 7.5 \cdot 10^{-9}$
Davies-Swann-Campey method with Gram-Schmidt orthogonalization	DSCG	75	$\varepsilon = 7.5 \cdot 10^{-9}$
Davies-Swann-Campey method with Palmer orthogonalization	DSCP	95	$\varepsilon = 7.5 \cdot 10^{-9}$
Powell's method of conjugate directions	POWE	135	$\varepsilon = 7.5 \cdot 10^{-9}$
Stewart's modification of the Davidon-Fletcher-Powell method	DFPS	180	$\varepsilon_a = \varepsilon_b = \varepsilon_c =$ $7.5 \cdot 10^{-9}$ [†]
Simplex Method of Nelder and Mead	SIMP	135	$\varepsilon = 10^{-8}$ [‡]
Method of Rosenbrock with Gram-Schmidt orthogonalization	ROSE	75	$\varepsilon = 10^{-4}$ [‡]
Complex method of Box	COMP	95	$\varepsilon = 10^{-6}$ [‡]
$(1+1)$ Evolution strategy	EVOL	4000	) $\varepsilon_a = \varepsilon_c =$) $3.0 \cdot 10^{-39}$
$(10, 100)$ Evolution strategy	GRUP	435	)
$(10, 100)$ Evolution strategy with recombination	REKO	435	) $\varepsilon_b = \varepsilon_d =$) $7.5 \cdot 10^{-9}$

[‡] Values fixed by the author.

[†] In place of the values set in Lill's program: $\varepsilon_a = 10^{-6}$, $\varepsilon_b = 10^{-10}$, $\varepsilon_c = 5 \cdot 10^{-13}$.

The maximum number of variables refers to an available core storage area of 30K words, which includes the main program and the FORTRAN library.

Besides their considerable cost in programming and computation time, numerical strategy comparisons entail further difficulties. The effectiveness of a method can be strongly influenced by small programming details. A number of methods were not fully worked out by their originators and require heuristic rules to be introduced before they can be applied. The way in which this degree of freedom is exercised to define the procedure depends on the skill and experience of the programmer, which leads to large discrepancies between the results of investigations and the judgements of different authors on one and the same strategy.

We have therefore, as far as possible, used already published programs (FORTRAN or ALGOL) for the algorithms or parts of them under study:

- One dimensional search with the Fibonacci method of Kiefer:

M. C. Pike, J. Pixner (1965)	Algorithm 2, Fibonacci search
J. Boothroyd (1965)	Certification of Algorithm 2
M. C. Pike, I. D. Hill, F. D. James (1967)	Note on Algorithm 2

- One dimensional search with the golden section method of Kiefer:

K. J. Overholt (1967)	Algorithm 16, Gold

- Direct search (pattern search) of Hooke and Jeeves:

A. F. Kaupe, Jr. (1963)	Algorithm 178, direct search
M. Bell, M. C. Pike (1966)	Remark on Algorithm 178
R. DeVogelaere (1968)	Remark on Algorithm 178
F. K. Tomlin, L. B. Smith (1969)	Remark on Algorithm 178
L. B. Smith (1969)	Remark on Algorithm 178

- Orthogonalization method for the strategies of Rosenbrock and of Davies, Swann, and Campey:

J. R. Palmer (1969)	An improved procedure for orthogonalizing the search vectors in Rosenbrock's and Swann's direct search optimization methods

- Derivative-free method of conjugate directions of M. J. D. Powell:

M. J. Hopper (1971)	Harwell subroutine library. A catalogue of subroutines, from which subroutine VA04A, updated May 20, 1970 (received as a card deck).

- Variable metric method of Davidon, Fletcher, and Powell as formulated by Stewart:

S. A. Lill (1970)	Algorithm 46. A modified Davidon method for finding the minimum of a function, using difference approximation for the derivatives.

S. A. Lill (1971)	Note on Algorithm 46
Z. Kovács (1971)	Note on Algorithm 46

Some of the parameters affecting the accuracy were altered, either because the small values defined by the author could not be realized on the available computer or because the closest possible approach to the objective could not have been achieved with them.

- Simplex method of Nelder and Mead:

R. O'Neill (1971)	Algorithm AS 47, function minimization using a simplex procedure

- A complete program for the Rosenbrock strategy:

M. Machura, A. Mulawa (1973)	Algorithm 450, Rosenbrock function minimization

This was not applied because it could only treat the unconstrained case.

- The same applies to the code for the complex method of M. J. Box:

J. A. Richardson, J. L. Kuester (1973)	Algorithm 454, the complex method for constrained optimization

The part of the strategy that, when the starting point is not feasible seeks a basis in the feasible region, is not considered here.

Whenever the procedures named were published in ALGOL they have been translated into FORTRAN. All the other optimization strategies not mentioned here have also been programmed in FORTRAN, with close reference to the original publications. If one wanted to repeat the test series today, a much larger number of codes could be made use of from the book of Moré and Wright (1993).

6.3.3 Results of the Tests

6.3.3.1 First Test: Convergence Rates for a Quadratic Objective Function

In the first part of the numerical strategy comparison the theoretical predictions of convergence rates and Q-properties will be tested, or, where these are not available, experimental data will be supplied instead. For this purpose two quadratic objective functions are used (Appendix A, Sect. A.1). In the first (Problem 1.1) the matrix of coefficients is diagonal with unit diagonal elements, i.e., a scalar matrix. This simplest of all quadratic problems is characterized by concentric contour lines or surfaces that can be represented or imagined as circles in the two parameter case, spheres in the three parameter case, and surfaces of hyperspheres in the general case. The same pattern of contours but with arbitrary monotonic variation in the objective function occurs in the sphere model for which the average rates of progress of the evolution strategies could be determined theoretically (Rechenberg, 1973; and Chap. 5 of this book).

The second objective function (Problem 1.2) has a matrix of coefficients with all non-zero elements. It represents a full quadratic problem (except for the missing linear term) with concentric, oblique ellipses, or ellipsoids as the contour lines or surfaces. The condition number of the matrix of coefficients increases quadratically with the number of parameters (see Appendix A, Sect. A.1). In general, the time required to calculate one value of the objective function increases as $O(n^2)$ for a quadratic problem, because, for a full form matrix, $\frac{n}{2}(n+1)$ distinct second order terms $a_{ij}\, x_i\, x_j$ must be evaluated. The objective function of Problem 1.2 has been formulated with the intention of reducing the computation time per function call to $O(n)$, without it being such a particular quadratic problem that one of the strategies could find it especially advantageous. The strategy comparison for this problem could thereby be made for much larger numbers of variables for the prescribed maximum computation time ($T_{max} = 8$ hours). The storage requirement for the full matrix A would also have been an obstacle to numerical tests with many parameters.

To enable comparison of the experimental and theoretical results, the required number of iterations, line searches, orthogonalizations, objective function calls, and the computation time were measured in going from the initial values

$$x_i^{(0)} = x_i^* + \frac{(-1)^i}{\sqrt{n}}, \qquad \text{for } i = 1(1)n$$

to an approximation

$$\left|x_i^{(k)} - x_i^*\right| \leq \frac{1}{10} \left|x_i^{(0)} - x_i^*\right|, \qquad \text{for } i = 1(1)n$$

The interval of uncertainty of the variables thus had to be reduced by at least 90%. The distance covered is effectively independent of the number of variables. The above conditions were tested after each iteration, and as soon as they were satisfied the search was terminated. The convergence criteria of the strategies themselves were not suppressed, but they could not generally take effect as they were much stricter. If they did actually operate it could be regarded as a failure of the method being applied.

The results of the first test are given in Tables 6.3 and 6.4. The number of function calls and the number of iterations or other characteristic processes involved are displayed in Figures 6.2 to 6.13 as a function of the number of parameters n on a log-log scale. As the data show, the computation time and effort of a strategy increase sharply with n. The large range in the number of variables compared to other investigations allows the trends to be seen clearly. To facilitate an overall view, the computation times of all the strategies are plotted as a function of the number of variables in Figures 6.14 and 6.15.

Table 6.3: Results of all strategies for test Problem 1.1

FIBO–Coordinate strategy with Fibonacci search

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	1	158	0.13
6	1	278	0.28
10	1	456	0.53
20	1	866	1.66
30	1	1242	3.07
60	1	2426	10.7
100	1	3870	26.5
200	1	7800	106
300	1	10562	210
600	1	21921	826
1000	1	38701	2500
2000	1	67451	8270
(max) 2900	1	103846	19300

1 cycle = n line searches

GOLD–Coordinate strategy with golden section

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	1	158	0.10
6	1	279	0.22
10	1	458	0.51
20	1	866	1.48
30	1	1242	3.14
60	1	2426	11.3
100	1	3870	27.6
200	1	7802	114
300	1	10562	221
600	1	21921	808
1000	1	38703	2670
2000	1	67431	8410
2900	1	103834	18300

1 cycle = n line searches

Table 6.3 (continued)

LAGR–Coordinate strategy with Lagrangian interpolation

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	1	85	0.04
6	1	163	0.12
10	1	271	0.30
20	1	521	0.88
30	1	781	1.80
60	1	1561	6.68
100	1	2501	17.3
200	1	5001	68.6
300	1	7201	153
600	1	14401	546
1000	1	24001	1620
2000	1	46803	6020
(max) 2540	1	64545	10300

1 cycle = n line searches

HOJE-Direct search of Hooke and Jeeves

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	4	20	0.02
6	4	43	0.04
10	3	48	0.06
20	7	274	0.50
30	3	168	0.43
60	8	874	3.70
100	2	352	2.37
200	8	3104	40.1
300	9	4954	100
600	7	7503	286
1000	12	23505	1460
2000	9	35003	4270
3000	10	58504	11200
(max) 4090	13	104300	25600

1 cycle = n to $2\,n$ individual steps

Table 6.3 (continued)

DSCG–Davies-Swann-Campey method with Gram-Schmidt orthogonalization

Number of variables	Number of orthog.	Number of line searches	Number of objective function calls	Computation time in seconds
3	0	3	20	0.04
6	0	6	34	0.10
10	0	10	56	0.20
20	0	20	111	0.68
30	0	30	136	1.18
50	0	50	226	2.80
(max) 75	0	75	338	6.10

DSCP–Davies-Swann-Campey method with Palmer orthogonalization

Number of variables	Number of orthog.	Number of line searches	Number of objective function calls	Computation time in seconds
(max) 95	0	95	428	9.49

Results for $n \leq 75$ identical to those of DSCG; in addition.

POWE–Powell's method of conjugate directions

Number of variables	Number of iterations	Number of line searches	Number of objective function calls	Computation time in seconds
3	1	3	11	0.02
6	1	6	20	0.06
10	1	10	32	0.12
20	1	20	62	0.32
30	1	30	92	0.60
60	1	60	182	1.96
100	1	100	202	3.72
(max) 135	1	135	407	8.60

1 complete iteration = $n + 1$ line searches; included are all the iterations begun

Table 6.3 (continued)

DFPS–Stewart's modification of the Davidon-Fletcher-Powell method

Number of variables	Number of iterations	Number of objective function calls	Computation time in seconds
3	1	10	0.02
6	1	16	0.04
10	1	24	0.06
20	1	44	0.16
30	1	64	0.32
60	1	124	1.14
100	1	204	3.19
135	1	274	5.42
(max) 180	1	364	9.56

1 iteration = 1 gradient evaluation and 1 line search

SIMP–Simplex method of Nelder and Mead (with restart)

Number of variables	Number of restarts	Number of objective function calls	Computation time in seconds
3	0	28	0.09
6	0	104	0.64
10	0	138	1.49
20	0	301	8.24
30	0	664	37.4
60	0	1482	277
100	0	1789	862
(max) 135	1	5142	5270

ROSE–Rosenbrock's method with Gram-Schmidt orthogonalization

Number of variables	Number of orthog.	Number of objective function calls	Computation time in seconds
3	1	27	0.08
6	2	60	0.32
10	2	120	0.91
20	1	181	2.56
30	0	121	1.18
40	1	281	13.7
50	2	550	48.4
60	2	600	78.3
(max) 75	2	899	145

Table 6.3 (continued)

COMP–Complex method of Box (2 n vertices)		
Number of variables	Number of objective function calls	Computation time in seconds
3	69	0.22
6	259	1.62
10	535	6.72
20	1447	72.0
30	2621	211
60	7263	2240
(max) 95	14902	11000

All numbers are averages over several attempts.

EVOL–(1+1) evolution strategy (average values)		
Number of variables	Number of mutations	Computation time in seconds
3	49	0.17
6	154	0.79
10	224	1.74
20	411	6.47
30	630	14.0
60	1335	60.0
100	2192	149
150	3322	340
200	4232	565
300	6666	1310
600	13819	5440
1000	23607	15600

Maximum number of variables (4,000) not achieved because too much computation time required.
Number of objective function calls = 1 + number of mutations

Table 6.3 (continued)

GRUP–(10 , 100) evolution strategy (average values)

Number of variables	Number of generations	Computation time in seconds
3	4	1.81
6	10	6.75
10	17	16.8
20	37	64.5
30	55	145
60	115	519
100	194	1600
200	377	5720
300	551	13600
(max) 435	854	28300

Number of objective function calls:
10 + 100 times number of generations.

REKO–(10 , 100) evolution strategy with recombination (average values)

Number of variables	Number of generations	Computation time in seconds
3	4	2.67
6	6	7.42
10	13	23.3
20	23	82.5
30	34	177
60	53	514
100	84	1420
200	136	4380
300	180	9340
(max) 435	289	21100

Number of objective function calls:
10 + 100 times number of generations.

Table 6.4: Results of all strategies for test Problem 1.2

FIBO–Coordinate strategy with Fibonacci search

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	8	928	0.68
6	22	4478	4.44
10	40	12644	15.6
20	87	50265	102
30	132	110423	298
50	227	297609	1290
60	282	422911	2120
100	Search terminates prematurely		

GOLD–Coordinate strategy with golden section

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	8	946	0.61
6	22	4418	3.96
10	40	12622	14.5
20	86	50131	102
30	133	111219	287
50	226	296570	1330
60	279	423471	2040
100	Search terminates prematurely		

LAGR–Coordinate strategy with Lagrangian interpolation

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	8	586	0.39
6	22	2826	2.48
10	40	8023	9.55
20	87	32452	62.8
30	134	70889	192
60	272	263067	1320
100	519	703130	5770
150	Search terminates prematurely		

Table 6.4 (continued)

HOJE–Direct search of Hooke and Jeeves

Number of variables	Number of cycles	Number of objective function calls	Computation time in seconds
3	11	65	0.04
6	30	353	0.34
10	26	502	0.62
20	78	3035	5.70
30	111	6443	16.3
60	212	24801	119
100	367	71345	547
200	727	284060	4270
300	1117	656113	14800

DSCG–Davies-Swann-Campey method with Gram-Schmidt orthogonalization

Number of variables	Number of orthog.	Number of line searches	Number of objective function calls	Computation time in seconds
3	3	16	87	0.22
6	7	55	195	0.87
10	8	101	323	2.70
20	16	361	1209	29.2
30	21	691	2181	110
40	42	1802	5883	484
50	27	1451	4453	582
60	44	2822	9308	1540
(max) 75	87	6676	20365	5790

DSCP–Davies-Swann-Campey method with Palmer orthogonalization

Number of variables	Number of orthog.	Number of line searches	Number of objective function calls	Computation time in seconds
3	3	16	84	0.22
6	7	55	194	0.78
10	8	101	324	1.54
20	16	361	1208	10.3
30	28	901	2809	33.8
50	28	1501	4610	89.7
75	79	6076	18591	547
(max) 95	100	9691	29415	1090

Table 6.4 (continued)

POWE–Powell's method of conjugate directions

Number of variables	Number of iterations	Number of line searches	Number of objective function calls	Computation time in seconds
3	3	11	27	0.08
6	5	35	77	0.30
10	9	99	215	0.97
20	17	354	744	4.82
30	53	1621	3401	24.1
40		search becomes infinite – no convergence		
50	175	8864	21532	235
60	138	8367	19677	222
70		search becomes infinite – no convergence		
80				
90				
100				
(max) 135				

DFPS–Stewart's modification of the Davidon-Fletcher-Powell method

Number of variables	Number of iterations	Number of objective function calls	Computation time in seconds	Fatal errors
3	3	20	0.04	
6	4	41	0.14	
10	5	74	0.34	
20	7	178	1.36	
30	9	333	3.63	
60	13	926	19.7	
100	17	2003	67.9	
135	20	3190	145	
(max) 180	22	4757	270	2 floating divide checks

Table 6.4 (continued)

SIMP–Simplex method of Nelder and Mead (with restart)

Number of variables	Number of restarts	Number of objective function calls	Computation time in seconds
3	0	29	0.09
6	1	173	1.06
10	0	304	3.17
20	0	2415	77.6
30	0	8972	579
40	2	28202	3030
50	1	53577	8870
60	1	62871	13700
70	1	86043	25800

ROSE–Rosenbrock's method with Gram-Schmidt orthogonalization

Number of variables	Number of orthog.	Number of objective function calls	Computation time in seconds
3	3	38	0.12
6	4	182	0.82
10	8	678	4.51
20	12	2763	35.6
30	14	5499	114
40	19	10891	329
50	21	15396	645
60	23	20911	1130
(max) 75	34	43670	3020

COMP–Complex method of Box ($2\,n$ vertices)

Number of variables	Number of objective function calls	Computation time in seconds
3	60	0.21
6	302	2.06
10	827	12.0
20	5503	235
30	24492	2330
	Search sometimes	terminates prematurely
40	Search always	terminates prematurely

All numbers are averages over several attempts

Table 6.4 (continued)

EVOL–(1+1) evolution strategy (average values)		
Number of variables	Number of mutations	Computation time in seconds
3	85	0.33
6	213	1.18
10	728	6.15
20	2874	44.4
30	5866	136
60	24089	963
100	69852	4690
150	152348	15200

GRUP–(10 , 100) evolution strategy (average values)		
Number of variables	Number of generations	Computation time in seconds
3	5	2.02
6	14	9.36
10	53	49.4
20	183	326
30	381	955
50	1083	4400
80	2977	18600
100	4464	35100

REKO–(10 , 100) evolution strategy with recombination (average values)		
Number of variables	Number of generations	Computation time in seconds
3	6	2.44
6	15	18.9
10	42	76.2
20	162	546
30	1322	6920
40	9206	61900

Figures 6.2 to 6.13 translate the numerical data into vivid graphics. The abbreviations used here are:

OFC	stands for objective function calls
ORT	stands for orthogonalizations

The parameters 1.1 and 1.2 refer to Problems 1.1 and 1.2 as mentioned above.

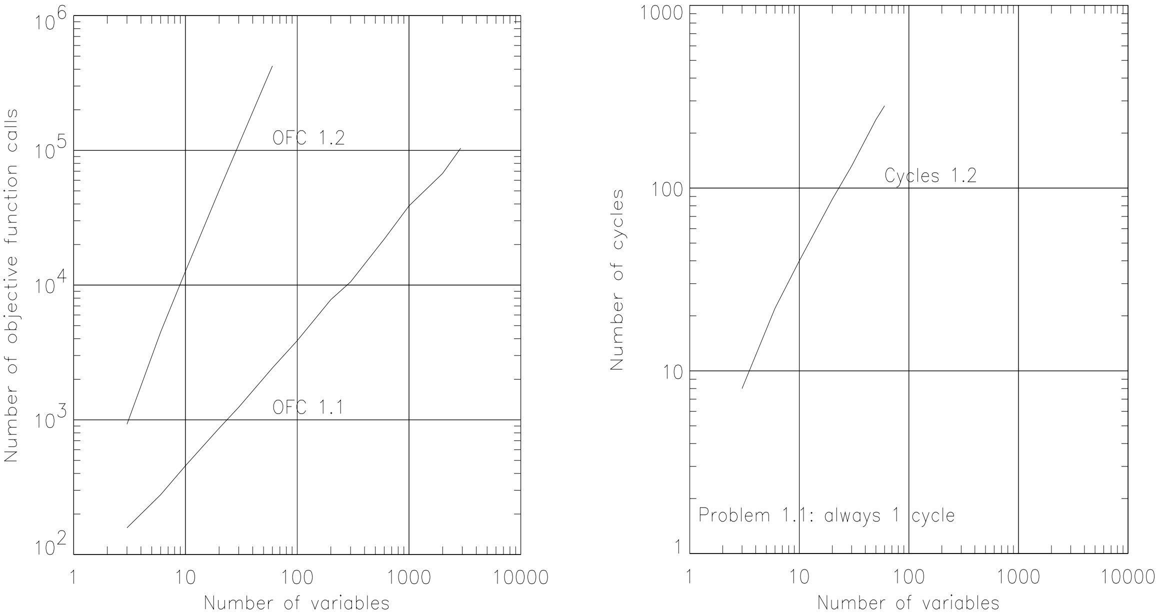

Figure 6.2: Coordinate strategy with Fibonacci search

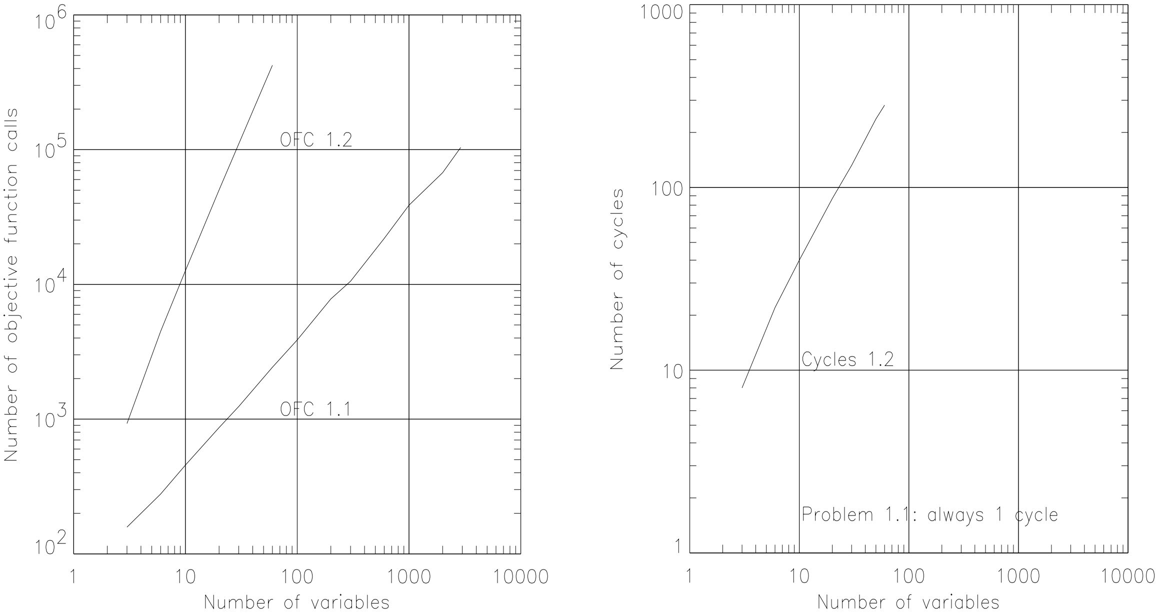

Figure 6.3: Coordinate strategy with golden section

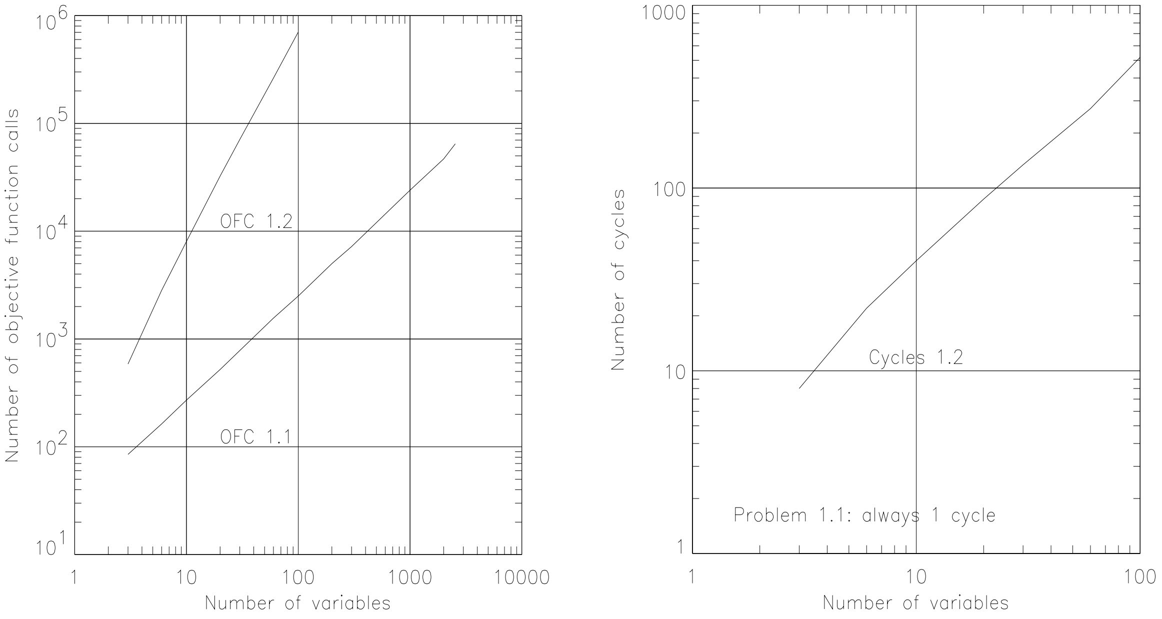

Figure 6.4: Coordinate strategy with Lagrangian interpolation

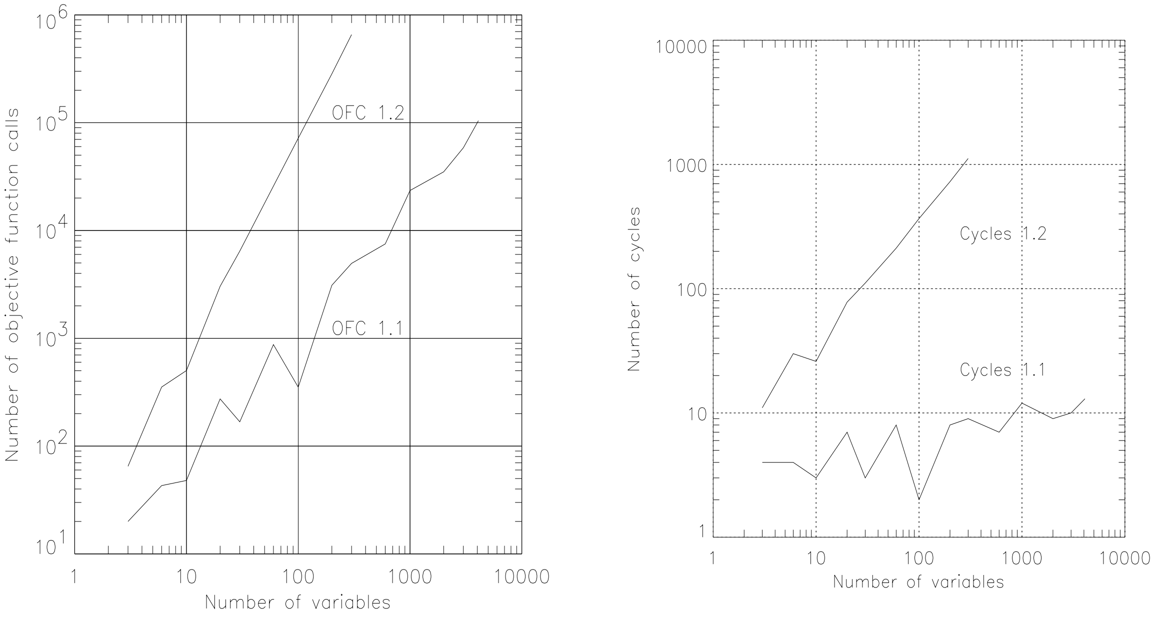

Figure 6.5: Strategy of Hooke and Jeeves

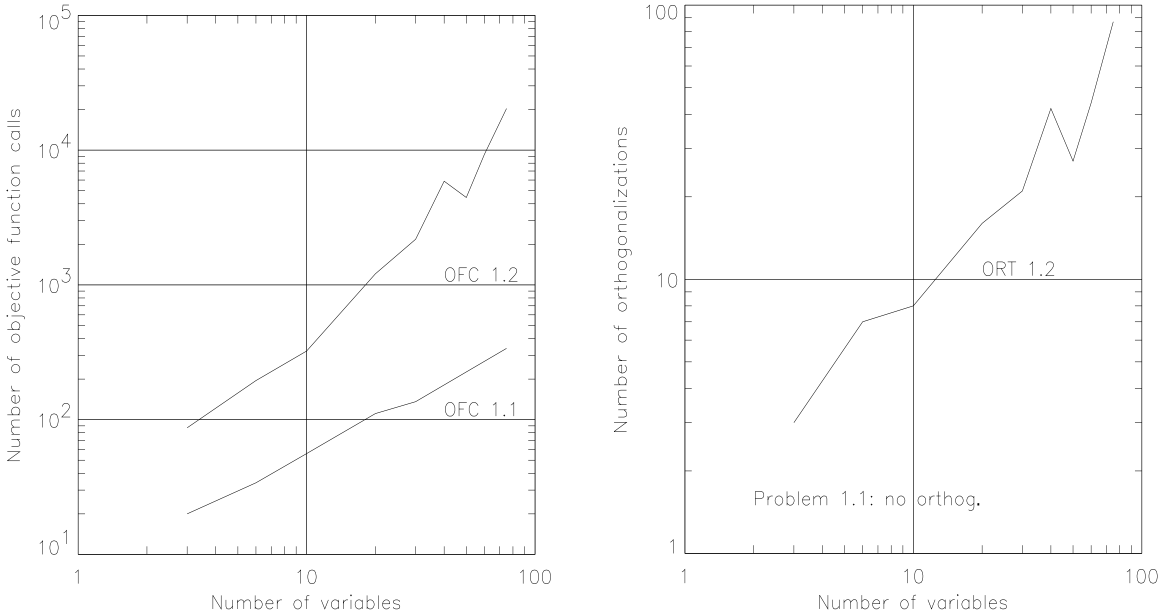

Figure 6.6: Strategy of Davies, Swann, and Campey with Gram-Schmidt orthogonalization

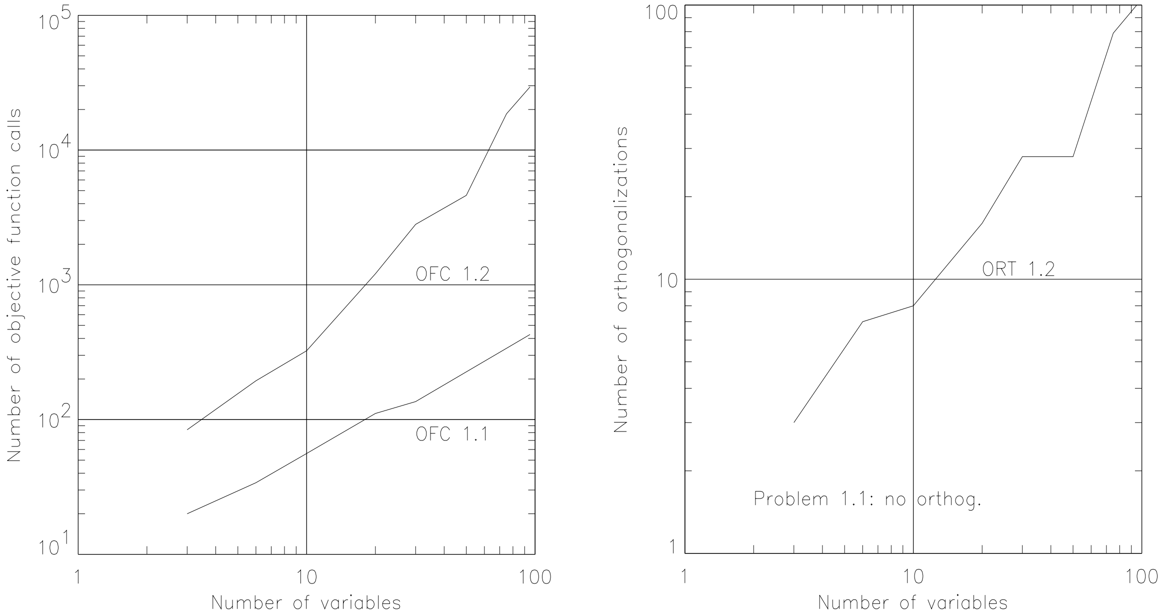

Figure 6.7: Strategy of Davies, Swann, and Campey with Palmer orthogonalization

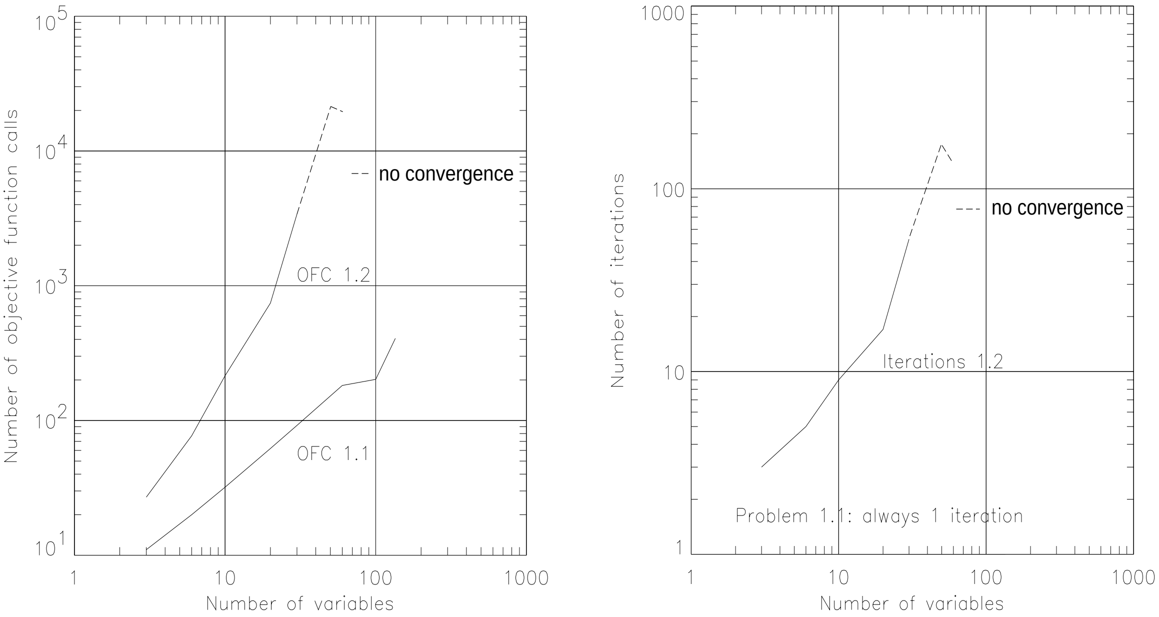

Figure 6.8: Strategy of Powell with conjugate directions

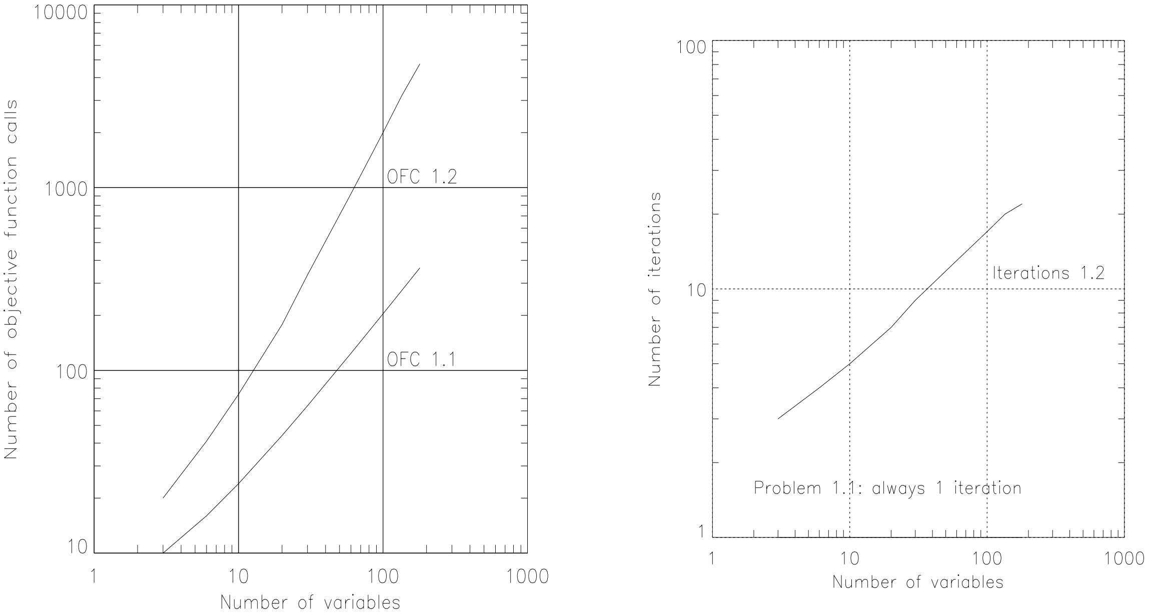

Figure 6.9: Strategy of Davidon, Fletcher, Powell, and Stewart as formulated by Lill (variable metric)

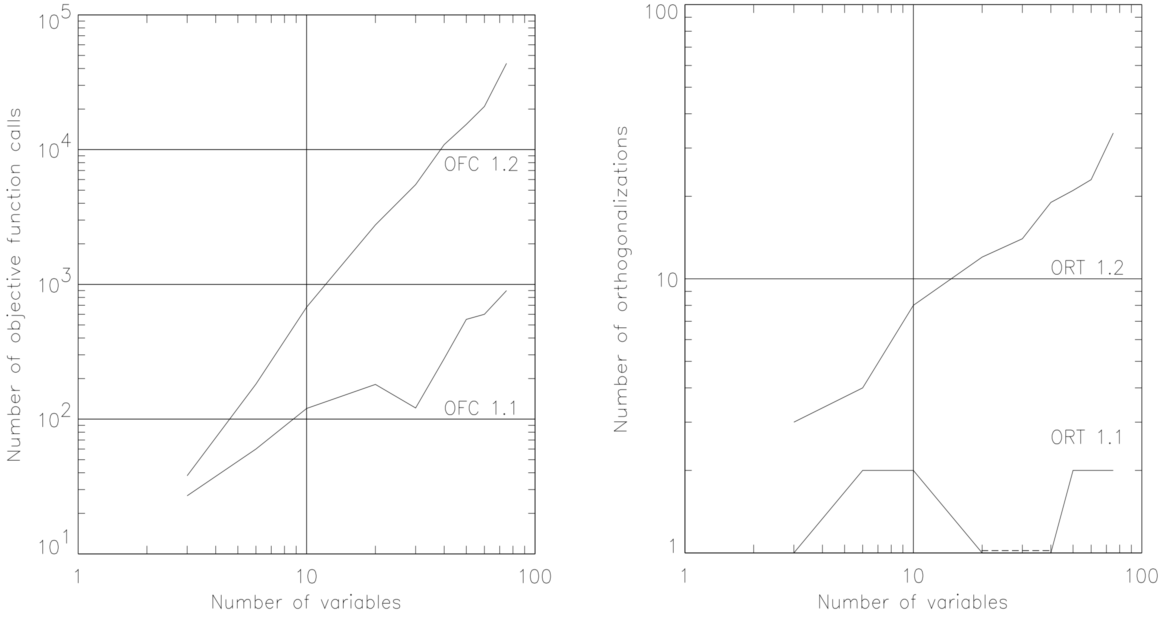

Figure 6.10: Strategy of Rosenbrock with Gram-Schmidt orthogonalization

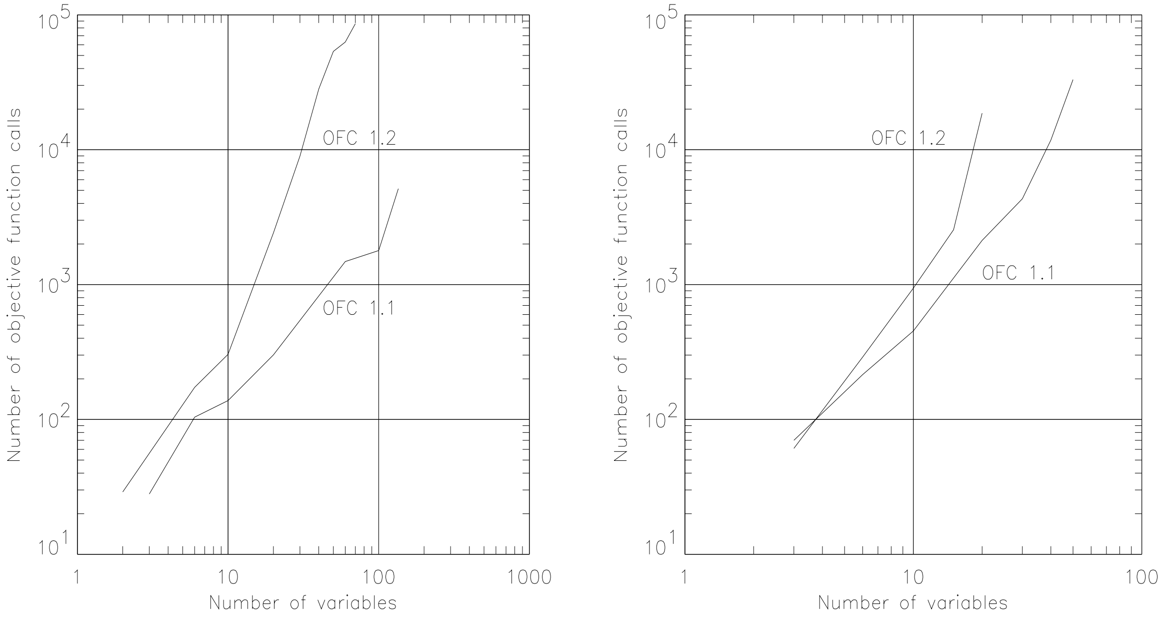

Figure 6.11: Left: Simplex strategy of Nelder and Mead,
Right: Complex strategy of Box

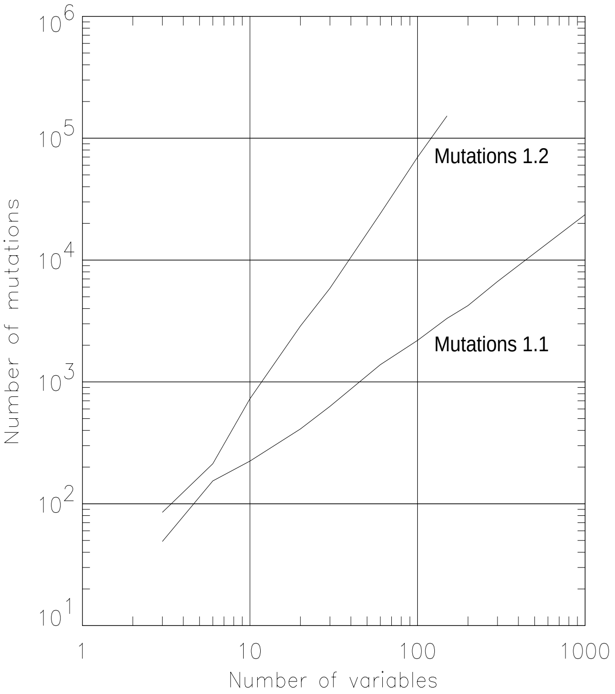

Figure 6.12: (1+1) evolution strategy

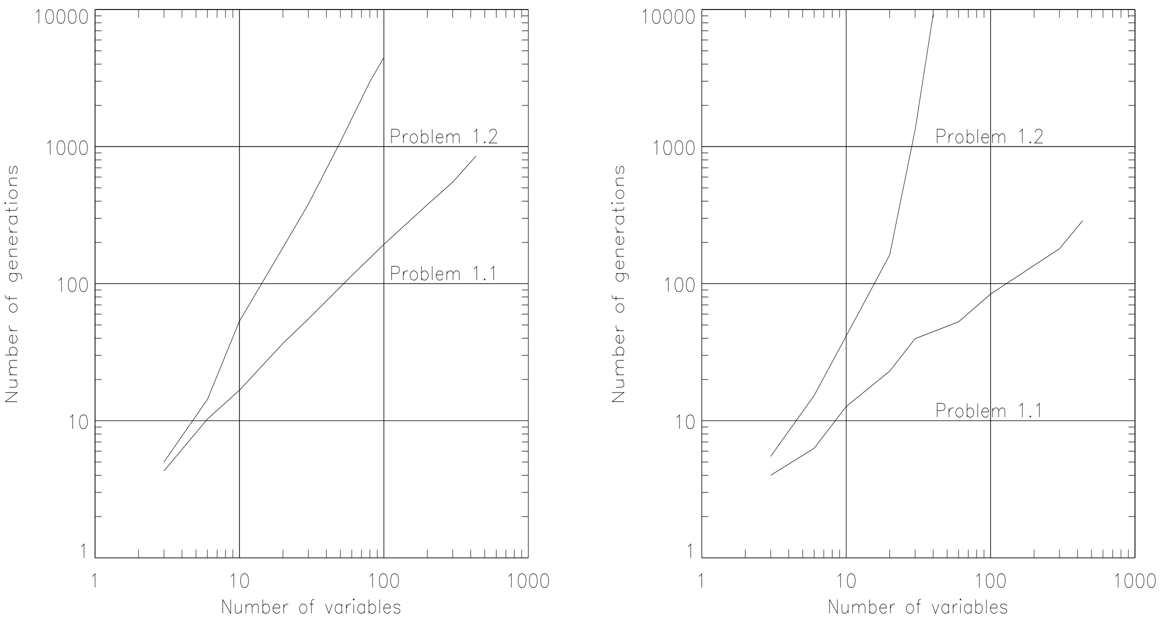

Figure 6.13: Left: (10 , 100) evolution strategy without recombination,
Right: (10 , 100) evolution strategy with recombination

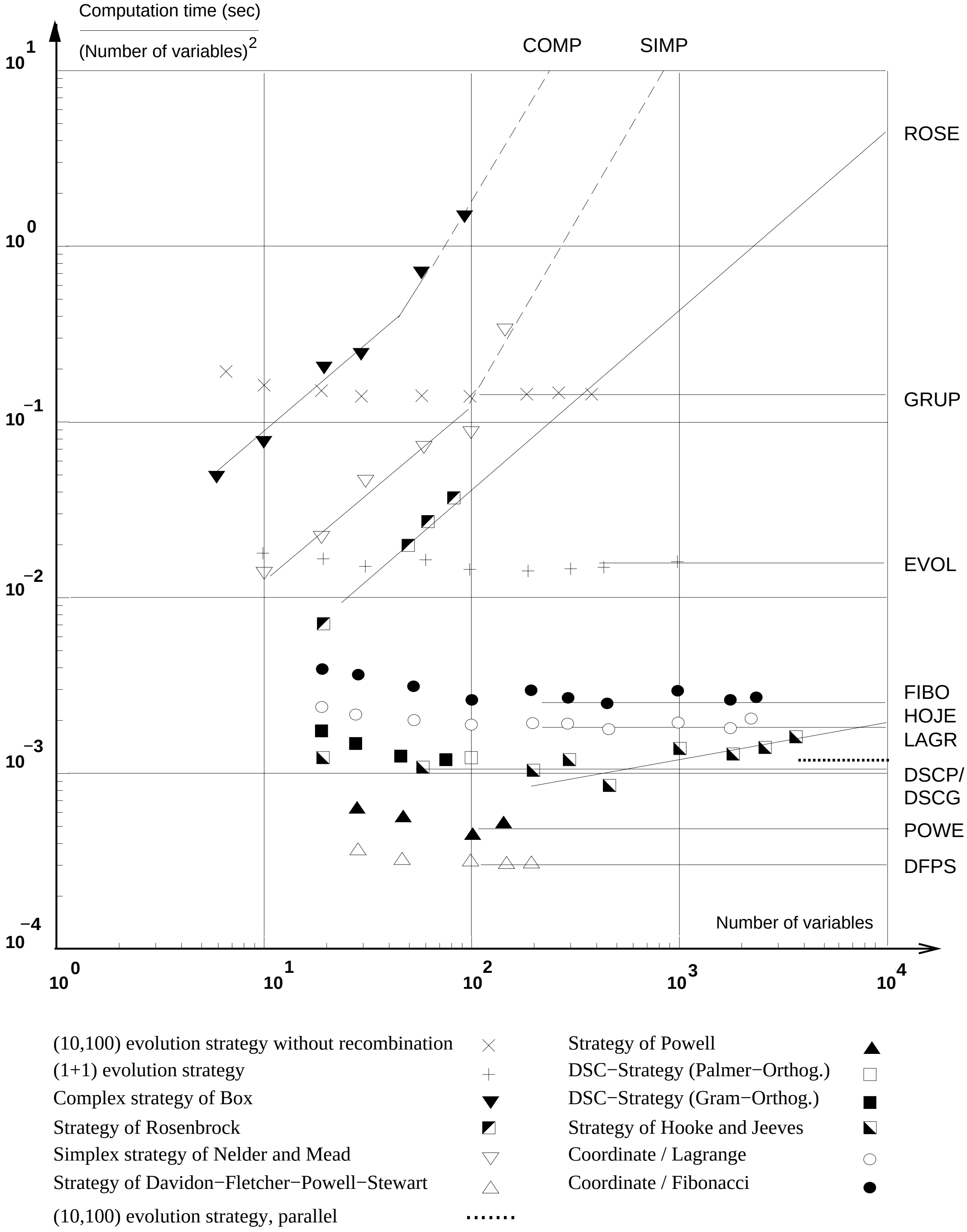

Figure 6.14: Result of the first comparison test: computation times for Problem 1.1

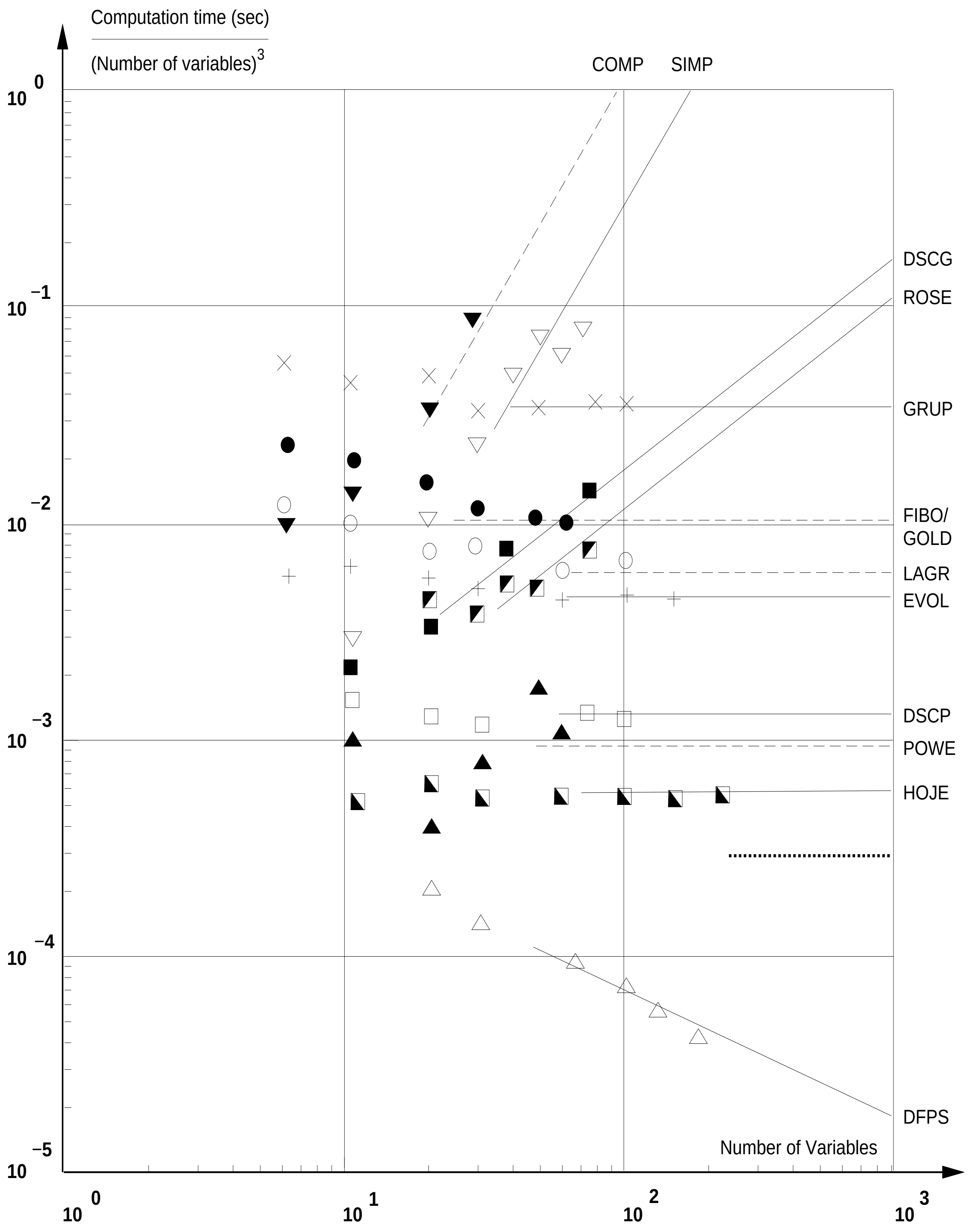

Figure 6.15: Result of the first comparison test:
computation times for Problem 1.2
Meanings of the symbols as in Figure 6.14

Points that deviated greatly from the trends have been omitted. To emphasize the differences between the methods, instead of the computation time T the quantities T/n^2 for Problem 1.1 and T/n^3 for Problem 1.2 have been plotted on a logarithmic scale.

For solving Problem 1.1 nearly all strategies require computation times of the order of $O(n^2)$. This corresponds to $O(n)$ objective function calls, each requiring $O(n)$ computation time. As expected, the most successful methods are the two that theoretically show quadratic convergence, namely the method of conjugate directions (Powell) and the variable metric method (DFPS). They obtain the solution within one iteration and n line searches respectively. For this simple problem, however, the same can be said for strategies with cyclic variation of the variables, since the search directions are the same. Of the three coordinate methods, the one with quadratic interpolation is a bit faster than the two which use sequential interval division. The latter two are of equal merit. The strategy of Davies, Swann, and Campey (DSC) also performs very well. Since the objective is reached within the first n line searches, no orthogonalizations need to be carried out. For this reason too both versions yield identical results for $n \leq 75$.

The evolution strategies live up to expectations in so far as the number of mutations or generations increases linearly with n. The number of objective function calls and the computation times are, however, considerably higher than those of the previously mentioned methods. For $r^{(0)}/r^{(M)} = 10$ the approximate theory of the two membered evolution strategy with optimal step length control predicts the number of mutations to be

$$M \simeq (5 \ln 10)\, n \simeq 11.5\, n$$

In fact nearly twice as many objective function calls (about $22\,n$) are required. This is partly because of the discrete way in which the variances are adjusted and partly because the chosen reduction factor of 0.85 corresponds to a success rate below the optimal value of 0.27. The ASSRS (adaptive step size random search) method of Schumer and Steiglitz (1968), which resembles the simple evolution strategy, is presently the most effective random method as far as we know. According to the experimental results of Schumer (1967) for Problem 1.2, taking into account the different initial and final conditions, it requires about the same number of steps as the (1+1) evolution strategy.

It is noteworthy that the (10 , 100) strategy without recombination only takes about 10 times as much time as the (1+1) method, in spite of having to execute 100 mutations per generation. This factor of acceleration is significantly higher than the theory for a (1 , 10) version would indicate and is closer to the calculated value for a (1 , 30) strategy. In the case of many variables, recombination further reduces the required number of generations by two thirds. This is less apparent in the computation time that is increased by the extra arithmetic operations, compared to the relatively inexpensive calculation of one objective function value. Thus, in the figures showing computation times only the (10 , 100) evolution without recombination has been included.

The strategy of Hooke and Jeeves appears to require computation times rather more than $O(n^2)$ on average; for many variables, nearer $O(n^{2.2})$. This arises from the slight increase with n of the number of exploratory moves. The likely cause is the fixed initial step length, which for problems with many variables is significantly too big and must first be reduced to the appropriate size. Three search strategies exhibit strikingly different behavior.

The method of Rosenbrock requires computation times on the order of $O(n^3)$. This can be readily understood. Up to the single exception of $n = 30$, in each case one or two orthogonalizations are performed. The Gram-Schmidt method employed performs $O(n^3)$ operations. If the number of variables is large the orthogonalization time is of major significance whenever the time for one function call increases less than quadratically with the number of variables. One can see here that the number of objective function calls is not always sufficient to characterize the cost of a strategy. In this case the DSC method succeeds with no orthogonalizations. The introduction of quadratic interpolation proves to give better results than the single step method of Rosenbrock.

Computation times for the simplex and complex strategies also increase as n^3, or even somewhat more steeply with n for many variables. The determining factor for the cost in this case is calculating the centroid of the simplex (or complex), about which the worst of the $(n + 1)$ or $2n$ vertices is reflected. This process takes $O(n^2)$ additions. Since the number of reflections and objective function calls increases as n, the cost increases, simply on this basis, as $O(n^3)$. Even in this simplest of all quadratic problems the simplex of the Nelder-Mead method collapses if the number of variables is large. To avoid premature termination of the optimum search, in the presently used algorithm for this strategy the simplex is initialized again. The search can thereby be prevented from stagnating in a subspace of $\mathbb{R}^n$, but the required computation time increases even more rapidly than $O(n^3)$. The situation is even worse for the complex method of Box. The author suggests using $2n$ vertices for problems with few variables and considers that this number could be reduced for many variables. However, the attempt to solve Problem 1.1 for $n = 30$ with a complex of 40 vertices fails in one of three cases with differing sequences of random numbers: i.e., the search process ends before achieving the required approximation to the objective. For $n = 40$ and 50 vertices the complex collapsed prematurely in all three attempts. With $2n$ vertices the complex strategy is successful up to the maximum possible number of variables, $n = 95$. Here again, however, for $n > 30$ the computation time increases faster than $O(n^3)$ with the number of parameters. It is therefore dubious whether the search would have been pursued to the point of reaching the maximum internally specified accuracy.

The second order methods only distinguish themselves from other strategies for solving Problem 1.1 in that their required computation time

$$T = c\,n^2, \qquad c = const.$$

is characterized by a small constant of proportionality c. Their full capabilities should become apparent in solving the true quadratic problem (Problem 1.2). The variable metric method lives up to this expectation. According to theory it has the property $Q\ n$, which means that after n iterations, n^2 line searches, and $O(n^3)$ computation time the problem should be solved. It comes as something of a surprise to find that the numerical tests indicate a requirement for only about $O(n^{0.5})$ iterations and $O(n^{2.5})$ computation time. This apparent discrepancy between theory and experiment is explained if we note that the property $Q\ n$ signifies absolute accuracy within at most n iterations, while in this example only a finite reduction of the uncertainty interval is required.

More surprising than the good results of the DFPS method is the behavior of the strategy of Powell, which in theory is also quadratically convergent. Not only does it require significantly more computation time, it even fails completely when the number of parameters is large. And in the case of $n = 40$ variables the step length goes to zero along a chosen direction. The convergence criterion is subsequently not satisfied and the search process becomes infinite; it must be interrupted externally. For $n = 50$ and $n = 60$ the Powell method does converge, but for $n = 70, 80, 90, 100$, and 130 it fails again. The origin of this behavior was not investigated further, but it may well have to do with the objection raised by Zangwill (1967) against the completeness of Powell's (1964) proof of convergence. It appears that rounding errors combined with small step lengths in the one dimensional search can cause linearly dependent directions to be generated. However, independence of the n directions is the precondition for them to be conjugate to each other.

The coordinate strategies also fail to converge when the number of variables in Problem 1.2 becomes very large. With the Fibonacci search and golden section as interval division methods they fail for $n \geq 100$, and with quadratic interpolation for $n \geq 150$. For successful line searching the step lengths would have to be smaller than allowed by the finite word length of the computer used. This phenomenon only occurs for many variables because the condition of the matrix of coefficients in Problem 1.2 varies as $O(n^2)$. In this proportion the elliptic contour surfaces $F(x) = const.$ become gradually more extended and the relative minimizations along the coordinate directions become less and less effective. This failure is typical of methods with variation of individual parameters and demonstrates how important it can be to choose other search directions. This is where random directions can prove advantageous (see Chap. 4).

Computation times for the method of Hooke and Jeeves and the method of Davies-Swann-Campey (DSC) clearly increase as $O(n^3)$ if Palmer orthogonalization is employed for the latter. For the method of Hooke and Jeeves this corresponds to $O(n)$ exploratory moves and $O(n^2)$ function calls; for the DSC method it corresponds to $O(n)$ orthogonalizations and $O(n^2)$ line searches and objective function evaluations. The original Gram-Schmidt procedure for constructing mutually orthogonal directions requires $O(n^3)$ rather than $O(n^2)$ arithmetic operations. Since the type of orthogonalization seems to hardly alter the sequence of iterations, with the Gram-Schmidt subroutine the DSC strategy takes $O(n^4)$ instead of $O(n^3)$ basic operations to solve Problem 1.2. For the same reason the Rosenbrock method requires computation times that increase as $O(n^4)$. It is, however, striking that the single step method (Rosenbrock) in conjunction with the suppression of orthogonalization until at least one successful step has been made in each direction requires less time than line searching, even if only one quadratic interpolation is performed. In both these methods the number of objective function calls, which is of order $O(n^2)$, plays only a secondary rôle.

Once again the simplex and complex strategies are the most expensive. From $n = 30$, the method of Nelder and Mead does not come within the required distance of the objective without restarts. Even for just six variables the search simplex has to be re-initialized once. The number of objective function calls increases approximately as $O(n^3)$, hence the computation time increases as $O(n^5)$. The strategy of Box with $2n$ vertices shows a correspondingly steep increase in the time with the number of variables. For $n = 30$

Problem 1.2 was actually only solved in one out of three attempts, and for $n = 40$ not at all. If the number of vertices of the complex is reduced to $n + 10$ the method fails from $n = 20$.

As in Problem 1.1, the cost of the evolution strategies increases rather smoothly with the number of parameters–more so than for several of the deterministic search methods. To solve Problem 1.2, $O(n^2)$ objective function calls are required, corresponding to $O(n^3)$ computation time. Since the distance to be covered is no greater than it was in Problem 1.1, the greater cost must have been caused by the locally smaller curvatures. These are related to the lengths of the semi-axes of the contour ellipsoids. Because of the regular structure of the matrix of coefficients A of the quadratic objective function in Problem 1.2, the condition number K, the ratio of greatest to least semi-axes (cf. test Problem 1.2 in Appendix A, Sect. A.1)

$$K = \left(\frac{a_{max}}{a_{min}}\right)^2$$

can be considered as the only quantity of significance in determining the geometry of the contour pattern. The remaining semi-axes will distribute themselves uniformly between a_{min} and a_{max}. The fact that K increases as $O(n^2)$ suggests that the rate of progress φ, the average change in the distance from the objective per mutation or generation, only decreases as the square root of the condition number. There is so far no theory for the general quadratic case. Such a theory will also look more complicated, since apart from the ratio of greatest to smallest semi-axis a further $n - 2$ parameters that determine the shape of the hyperellipsoid will play a rôle. The position of the starting point will also have an effect, although in the case of many variables only at the beginning of the search. After a transition phase the starting point of mutations will always lie in the vicinity of a point where the objective function contour surfaces are most curved. In the sphere model theory of Rechenberg, if r is regarded as the average local radius of curvature, the rate of progress at worst should become inversely proportional to the square root of the condition number. The convergence rate of the evolution strategy would then be comparable to that of the strategy of steepest descents, for which function values of two consecutive iterations in the quadratic case are in the ratio (Akaike, 1960)

$$\frac{a_{max} - a_{min}}{a_{max} + a_{min}}$$

Compared to other methods having costs in computation time that increase as $O(n^3)$, the evolution strategies fare better than they did in Problem 1.1. Besides the fact that the coordinate strategies do not converge at all when the number of variables becomes large, they are surpassed in speed by the two membered evolution strategy. The relative performance of the two membered and multimembered evolution strategies without recombination remains about the same.

The behavior of the (10 , 100) evolution strategy with recombination deviates from that of the other versions. It requires considerably more computation time to solve Problem 1.2. This can be attributed to the fact that, although the probability distribution for mutation steps alters, it cannot adapt continuously to the local conditions. Whilst the mutation ellipsoid, the locus of all equiprobable mutation steps, can extend and contract along the coordinate directions, it cannot rotate in the space. To do so, not only the variances but also the orientation or covariances would need to be variable (for such an extension see Chap. 7 and subroutine KORR). As the results show, starting from a spherical shape the mutation ellipsoid adopts a configuration that initially accelerates the search process. As it progresses towards the objective the ellipsoid must become smaller; but it should also gradually rotate to follow the orientation of the contour lines. That is not possible because the mechanism adopted here allows no mutual dependence of the components of the random vector. The ellipsoid first has to form itself into a sphere again, or to become generally small, before it extends again with the longer axes in new directions. This awkward process actually occurs, but it causes an appreciable delay in the search.

There is a further undesirable phenomenon. Supposing that a single variance suddenly becomes very much smaller. The associated variation in the variables then takes place in an $(n - 1)$-dimensional subspace of $\mathbb{R}^n$ (for a more detailed analysis see Schwefel, 1987). Other things being equal, the probability of a success is thereby greater than if all the parameters had varied. Step length alterations of this kind are therefore favored and, together with the resistance to rotation of the mutation ellipsoid, they enhance the unstable behavior of the strategy with recombination. This can be prevented by having a large population, in which there is always a sufficient supply of different kinds of parameter combinations for the variances as well. Another possibility is to allow one individual to execute several consecutive mutations with one setting of the step length parameters. Then the overall success depends rather less on the instantaneous probability of success and more on the size of the partial successes. The quality of the strategy parameters is thereby assessed more objectively. It should be noticed that Problem 1.2 is actually the only one in which recombination appears troublesome. In many other cases it led to a reduction in the computation cost, even in the simple form applied here (see second and third test).

6.3.3.2 Second Test: Reliability

Convergence in the quadratic case is a minimum requirement of non-linear optimization methods. The unsatisfactory results of the coordinate strategies and of Powell's method for a large number of variables confirm the necessity of numerical tests even when convergence is assured by theory. Even more important, in fact unavoidable, are experimental tests of the reliability of convergence of optimization methods on non-quadratic, non-linear problems. Some methods with an internal quadratic model of the objective function have to be modified in order to deal with more general problems. Such, for example, is the method of conjugate gradients. The method of Fletcher and Reeves (1964) actually terminates after the relative minimum has been obtained in each of n conjugate directions.

However, for higher order objective functions the optimum will not have been reached after n iterations. Even in quadratic problems, if they are ill-conditioned, more iterations may be required. There are two possible ways to proceed. Either the iteration process can be formally continued beyond n line searches or it can be repeated in a cyclic way. Fletcher and Reeves recommend destroying all the accumulated information after each set of $n + 1$

iterations and beginning again, i.e., with uncorrected gradient directions. This procedure is said to be more effective for non-quadratic objective functions. On the other hand, Fox (1971) suggests that a periodic restart of the search can prevent convergence in the quadratic case, whereas a simple continuation of the sequence of iterations is successful. Further suggestions for the way to restart are made by Fletcher (1972a).

The situation is similar for the quasi-Newton methods in which the Hessian matrix or its inverse is approximated in discrete steps. Some of the proposed formulae for improving the approximation matrix can lead to division by zero; sometimes due to rounding errors (Broyden, 1972), but in other cases even on theoretical grounds. If the Hessian matrix has singular points, the optimization process stagnates before reaching the optimum. Bard (1968) and others recommend as a remedy replacing the approximation matrix from time to time by the unit matrix. The information gathered over the course of the iterations is destroyed again in this process. Pearson (1969) proposes a restart period of $2\,n$ cycles, while Powell (1970b) suggests regularly adding steps different from the predicted ones. It is thus still true to say of the property of quadratic termination that its "relevance for general functions has always been questionable" (Fletcher, 1970b). No guarantee is given that Newtonian directions are better than the (anti-) gradient.

As there is no single objective function that can be taken as representative for determining experimentally the properties of a strategy in the non-quadratic case, as large and as varied a range of problem types as possible must be included in the numerical tests. To a certain extent, it is true to say that the greater their number and the more skillfully they are chosen, the greater the value of strategy comparisons. Some problems have become established as standard examples, others are added to each experimenter's own taste. Thus in the catalogue of problems for the second series of tests in the present strategy comparison, both familiar and new problems can be found; the latter were mainly constructed in order to demonstrate the limits of usefulness of the evolution strategies.

It appears that all the previously published tests use as a basis for judging performance the number of function calls (with objective function, gradient, and Hessian matrix weighted in the ratio $1 : n : \frac{n}{2}(n+1)$) and the computation time for achieving a prescribed accuracy. Usually the objective functions considered are several times continuously differentiable and depend on relatively few variables, and the results lack compatibility from problem to problem and from strategy to strategy. With one method, a first minimum may be found very quickly, and a second much more slowly; another method may work just the opposite way round. The abundance of individual results actually makes a comprehensive judgement more difficult. Hence average values are frequently calculated for the required computation time and the number of function calls. Such tests then result in establishing that second order methods are faster than first order and these in turn are faster than direct search methods. These conclusions, which are compatible with the test results for quadratic problems, lead one to suspect that the selected objective functions behave quadratically, at least in the neighborhood of the objective. Thus it is also frequently noted that, at the beginning of a search, gradient methods converge faster, whereas towards the end Newton methods are faster. The average values that are measured therefore depend on the chosen starting point and the required closeness of approach to the objective.

The assessment is tricky if a method does not converge for a particular problem but terminates the search following its own criteria without getting anywhere near the solution. Any strategy that fails frequently in this way cannot be recommended for use in practice even if it is especially fast in other cases. In a practical problem, unlike a test problem, the correct solution is not, of course, known in advance. One therefore has to be able to rely on the results given by a strategy if they cannot be checked by another method. Hence, reliability is just as important a criterion for assessing optimization methods as speed.

The second part of the strategy comparison is therefore designed to test the robustness of the optimization methods. The scale for assessing this is the number of problems that are solved by a given method. Since in this respect it is the complexity rather than size of the problem that is significant, the number of variables ranges only from one to six.

All numerical iteration methods in practice can only approximate a solution with a finite accuracy. In order to be able either to accept the end result of an optimum search as adequate, or to reject it as inadequate, a border must be defined explicitly, on one side of which the solution is exact enough and on the other side of which it is unsatisfactory. It is the structure of the objective function that is the decisive factor determining the accuracy that can be achieved (Hyslop, 1972). With this in mind the border values for the purpose of ranking the test results were obtained by the following scheme. Starting from the known exact or best solution

$$x^* = (\, x_1^*,\ x_2^*,\ \ldots,\ x_n^* \,)^T$$

the variables were individually altered by the amounts

$$\triangle x_i = \begin{cases} \pm\delta, & \text{for } x_i^* = 0 \\ \pm x_i^*\,\delta, & \text{for } x_i^* \neq 0 \end{cases}$$

in all combinations. For example for $n = 2$ one obtains eight different test values of the objective function (see Fig. 6.16). In the general case there are $3^n - 1$ different values. The greatest deviation $\triangle F(\delta)$ from the optimal value $F(x^*)$ defines the border between results that approach the objective sufficiently closely and results that do not. To obtain a number of grades of merit, four different test increments δ_j, $j = 1(1)4$ were selected:

$$\begin{aligned} \delta_1 &= 10^{-38} \\ \delta_2 &= 10^{-8} \\ \delta_3 &= 10^{-4} \\ \delta_4 &= 10^{-2} \end{aligned}$$

A problem is deemed to have been solved "exactly" at $\tilde{x}$ if

$$F(\tilde{x}) \leq F(x^*) + \triangle F(\delta_1)$$

is attained. On the other hand, if at the end of the search

$$F(\tilde{x}) > F(x^*) + \triangle F(\delta_4)$$

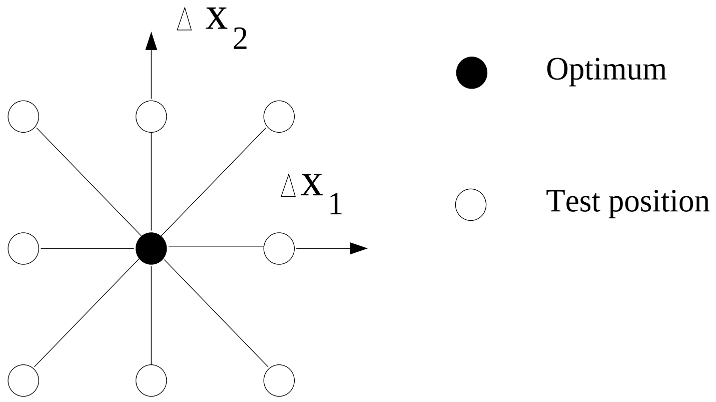

Figure 6.16: Eight different test values of the objective function in case of $n = 2$

the strategy employed has failed. Three intermediate classes of approximation are defined in the obvious way.

The maximum possible accuracy was required of all strategies. The corresponding free parameters of the strategies that enter the termination criteria have already been defined in Table 6.2. In contrast to the first test, no additional common termination rule was employed.

A total of 50 problems were to be solved. The mathematical formulations of the problems are given in Appendix A, Section A.2. Some of them are only distinguished by the chosen initial conditions, others by the applied constraints. Nine out of 14 strategies or versions of basic strategies are not suited to solving constrained problems, at least not directly. Methods involving transformations of the variables and penalty function methods were not employed. An exception is the method of Rosenbrock, which only alters the objective function near the boundaries and can be applied in one pass; otherwise penalty functions require a sequence of partial optimizations to be executed. The second series of tests therefore comprises one set of 28 unconstrained problems for all 14 strategies and a second set of 22 constrained problems for 5 of the strategies. The results are displayed together in Tables 6.5 to 6.8. The approximation to the objective that has been achieved in each case is indicated by a corresponding symbol, using the classes of accuracy defined above.

Any interesting features in the solution of individual problems are documented in the Appendix A, Section A.2, in some cases together with a brief analysis. Thus at this point it is only necessary to make some general observations about the reliability of the search methods for the totality of problems.

Unconstrained Problems

The results of the three versions of the coordinate strategies are very similar and generally unsatisfactory. A third of all the problems cannot be solved with them at all, or only very inaccurately. Exact solutions ($\delta = 10^{-38}$) are the exception and only in less than a

Table 6.5: Results of all strategies in the second comparison test, unconstrained problems

Problem	F	G	L	H	D	D	P	D	S	R	C	E	G	R
	I	O	A	O	S	S	O	F	I	O	O	V	R	E
	B	L	G	J	C	C	W	P	M	S	M	O	U	K
No.	O	D	R	E	G	P	E	S	P	E	P	L	P	O
2.1	3	3	3	1	1	1	2	1e	2n	1	2	1	1	1
2.2	3	3	3	1	1	1	1	2	2a	1a	2	1	1	1
2.3	2	2	3	1	1	1	1	1	1n	1a	4	1	1	1
2.4	3	3	3	1	1	1	2	2e	3	1	3	1	1	1
2.5	2	2	2	2	1	1	2	1e	2a	1	2	1	1	1
2.6	5	5	5	5	2	2	5	3	2a	1	3	1	1	1
2.7	5	5	4	2	5ea	5ea	5e	5	3	5	3	2	1	1
2.8	5	5	4	3	5ea	5ea	5e	5	3	5	2	1	1	1
2.9	3	3	3	2	2	2	2e	2	1a	3	1	1	2	1
2.10	5	5	4	3	2	2	5a	4e	4n	2	2	4	3	1
2.11	5	5	5	3	2	2	2	4	4n	2	2	4	3	1a
2.12	5	5	5	3	2	3	2	4e	2	2	2	4	3	1a
2.13	3	3	3	2	2	1	3	2	3	3	3	2	1	1
2.14	3	3	3	2	2	2	2a	5e	2n	2	2	3r	3r	3r
2.15	3	3	3	2	2	2	2ea	3	2	2	2	3r	3r	3r
2.16	2	1	2	2	1	1	2	2	2n	2	3	3	2	2
2.17	2	2	1	2	2	2	2e	2	1a	2	1	1	1	1
2.18	5	5	5	2	2	2	2e	2	1an	1	1	1	1	1
2.19	5	5	5	5	2	2	5	2e	2	2	3	3	2	3
2.20	2	2	2	2	3	2	2	1e	3n	2	2	1	1	1
2.21	5	5	5	2	4	2	5	2e	5a	2	5	1	1	1
2.22	2	2	2	2	2	2	2	5	5a	2	5	1	1	1
2.23	1	1	1	1	1	1	5a	5e	1a	1a	1	1	1	1
2.24	3	3	3	2	1	1	2	2	2	2	2	2	2	1
2.25	3	3	3	2	1	1	2e	2e	3	1	3	1	1	1
2.26	1	1	2	1	1	1	1e	1	1n	1	4	1	1	1
2.27	1	1	5	2	1	1	1	5	1	1	2	1	1	1
2.28	4	4	4	3	4	3	2	4e	2	3	1	4	4	3
Sum	91	90	93	61	56	52	74	79	65	54	68	51	51	37

Meaning of the number and letter symbols used above:

1 Accuracy achieved better than 10^{-38}
2 Accuracy achieved better than 10^{-8}
3 Accuracy achieved better than 10^{-4}
4 Accuracy achieved better than 10^{-2}
5 Accuracy achieved worse than 10^{-2}

e Fatal execution error (floating overflow, floating divide check)
a Termination rule ineffective; search infinite with no further convergence
r Computation time too long or convergence too slow; search terminated
n Concerns the simplex method of Nelder and Mead: restart(s) required

third of all cases are the end results good ($\delta \leq 10^{-8}$). As already shown by the quadratic objective function models, it appears again that progress along the directions of the unit vectors becomes possible only in very small step lengths. The limit of smallest possible changes in the variables, as defined by the finite word and mantissa lengths of the digital computer, is often reached before the search has come sufficiently close to the objective.

The three methods with rotating axes also behave similarly to one another, namely the strategies of Rosenbrock and of Davies, Swann, and Campey. Although the choice of orthogonalization method (Gram-Schmidt or Palmer) has a considerable effect on the computation times it makes little difference to the accuracies achieved. If "exact" solutions are required, all three methods prove useful in about 4 out of 10 cases. This proportion is doubled if the accuracy requirement is lowered by a grade. Two problems (Problems 2.7 and 2.8) are not solved by any of the three variants. In the Rosenbrock method, the search is ended a very long way from the objective, while in the DSC method a line search becomes infinite. To prepare for the single quadratic interpolation it uses a subroutine for bounding the relative minimum in the chosen direction. In this case, however, the relative minimum is situated at infinity; thus, after some time, the range of numbers that can be handled by the computer is exceeded. It eventually makes a fatal execution error with the message: "floating overflow." In most computers, a program would terminate at this point, but the PDP 10 continues the calculation using its largest number 2^{127} in place of the value that exceeded the number range. Nevertheless the bounding procedure does not end because in the DSC method any steps that do not change the value of the objective function are also regarded as successful. The convergence criterion is not tested within this subroutine, so the whole procedure becomes infinite without any further change in value of the objective function. It must be terminated externally. The convergence criterion of the Rosenbrock method fails in three cases, in spite of the fact that the exact solutions have already been found. It is noted on the tables wherever fatal execution errors occur or the optimization does not terminate normally. With 11 or 12 exact results, and altogether 23 good results, these three rotating axes methods rank highly.

Fatal errors occur especially frequently in applying the more "thoroughbred" methods, the method of Powell and the DFPS strategy. They are not always accompanied by termination difficulties or bad final results. The accuracies achieved have therefore been evaluated independently of the execution errors. Good approximations, of which there are 20 (Powell) and 16 (DFPS) out of 28, are also less frequent than in the orthogonalization strategies. In many cases both of these methods that are so advantageous in theory completely fail to approach the desired solution; usually in the same problems that present difficulties with the much simpler coordinate methods.

Apart from failure of a line search because of a relative minimum at infinity, the causes are:

- The confusion of minima and saddle points because of ambiguity in quadratic interpolation (Problem 2.19 for the Powell strategy, Problem 2.27 for the variable metric method)

- Discontinuities in the objective function or its derivatives (Problems 2.6, 2.21, 2.22)

- A singular Hessian matrix (Problem 2.14 in the DFPS method)

However, even a completely regular, several times differentiable objective function of 10th order (Problem 2.23) is not managed by either of the quadratically convergent strategies. Their concept of using all the data that can be accumulated during the iterations to adjust their internal quadratic model apparently leads to completely wrong predictions of favorable directions and step lengths if the function is of appreciably higher than second order. Not one of the other direct search methods fails on this problem; in fact they all find the exact solution.

With Powell's method one can choose between two different convergence criteria. The difference between the stricter one and the simple one is that the former displaces slightly the best position obtained after the sequence of iterations has ended normally and searches again for the minimum. The search is only finally terminated if both results are the same within the specified accuracy. Otherwise the search is continued after a line search in the direction of the difference vector between the two solutions. Because of the extreme accuracy requirements in the present cases the search usually ends with the message that rounding errors in the objective function prevent any closer approach to the objective. In such cases no additional variation in the final result is made. Even in other cases, the stricter convergence criterion only makes a very slight improvement of the results; the grades of merit of the results are not changed at all. In four problems the search becomes infinite because the step lengths vanish and the termination criterion is no longer tested. The search has to be terminated externally. Fatal execution errors occur very frequently. In three cases there is a "floating overflow" and in seven cases a "floating divide check." This concerns a total of eight problems. The DFPS strategy is even more susceptible. There are five occurrences of "floating overflow" and eleven of "floating divide check." Twelve problems are involved.

In contrast, the direct search of Hooke and Jeeves works without errors, but even this method fails on two problems; one because of sharp corners in the pattern of contour lines (Problem 2.6) and another in the neighborhood of a stationary point with a very narrow valley leading to the objective (Problem 2.19). Nevertheless it yields 6 exact solutions and 21 good approximations.

The overall behavior of the simplex and complex strategies is similar, but there are differences in detail. There are 17 good solutions together with 6 exact ones to set against two failures (Problems 2.21 and 2.22). These are provoked by edges on the contour surfaces in the multidimensional space. The restart rule in the Nelder-Mead method is invoked during 9 of the solutions. The termination criterion based only on function values at the simplex corners does not operate in 9 cases. The optimum search becomes infinite with no apparent improvement in the objective function values. The results of the complex strategy depend strongly on the initial configuration, which is determined by random numbers. In this case the evaluation was made for the best of three attempts each with different sequences of pseudorandom numbers. It is especially worth noting the performance of the complex method in solving Problem 2.28, for which it is better than all the other methods.

All three versions of the evolution strategy are distinguished by the fact that in no case do they completely fail, and they are able to solve far more than half of all the problems exactly (in the sense defined above). Since their behavior, like that of the complex method, is influenced by random numbers, the same rule was followed: namely, out of three tests the one with the best end result was accepted. In contrast to the strategy of Box, however, the evolution methods prove to be less dependent on the actual sequence of random numbers. This is especially true of the multimembered versions. Recombination almost always improves the chance of getting very close to the desired solutions. Fatal errors due to exceeding the maximum number range or dividing by zero do not occur by virtue of the simple computational operations in these strategies. Discontinuities in the partial derivatives, saddle points, and the like have no obvious adverse effects. The search does, however, become rather time consuming when the minimum is reached via a long, narrow valley. The step lengths or variances that are set in this case are very small and impose slow convergence in comparison to methods that can perform a line search along the valley. The average rate of progress of an evolution strategy is not, however, affected by bends in the valley, which would retard a one dimensional minimization procedure. Line searches only afford a significant advantage to the rate of progress if there are directions in the space along which successful steps can be made of a size that is large compared to the local radius of curvature of the objective function contour surface. Examples are provided by Problems 2.14, 2.15, and 2.28. In these cases, long before reaching the minimum the optimal variances of the evolution methods have reached the lower limit as determined by the machine accuracy. The desired solution cannot therefore be approximated to the required accuracy. In Problems 2.14 and 2.15 the computation time limit did not allow the convergence criterion to be satisfied; although it was actually progressing slowly but surely, the search was terminated.

Difficulties with the termination rule based on function values only occurred in the solution of one type of problem (Problems 2.11, 2.12) using the (10 , 100) evolution strategy with recombination. The multimembered method selects the 10 best individuals of a generation only from the current 100 descendants. Their 10 parents are not included in the selection process, for reasons associated with the step length adaptation. In general, the objective function value of the best descendant is closer to the solution than that of the best parent. In the case of the two problems referred to above, this is initially the case. As the solution is approached, however, it happens more and more frequently that the best value occurring in a generation is lost again. This is related to the fact that because of rounding errors in evaluating values near the minimum, the objective function behaves practically stochastically. Thus the population wanders around in the neighborhood of the (quasi-singular) optimal solution without being able to satisfy the convergence criterion. These difficulties do not beset the other search methods, including the multimembered evolution without recombination, because they do not come nearly so close to the optimum. The fact that the third problem of the same type (Problem 2.10) is solved without difficulties in a finite time, even with recombination, can be considered a fluke. Here too the minimum was reached long before the termination criterion was satisfied. On the whole, the multimembered evolution strategy with recombination is the surest and safest of all the search methods tested. In only 5 out of 28 cases is the solution not located exactly, and the greatest deviations of the variables were in the accuracy class $\delta = 10^{-4}$.

Table 6.6: Summary of the results from Table 6.5

Strategy	Total number of problems solved in the accuracy class $\delta \leq$				No solution or $\delta > 10^{-2}$	Fatal computation errors	No normal termination
	10^{-38}	10^{-8}	10^{-4}	10^{-2}			
FIBO	3	9	18	19	9	0	0
GOLD	4	9	18	19	9	0	0
LAGR	2	7	17	21	7	0	0
HOJE	6	21	26	26	2	0	0
DSCG	11	23	24	26	2	2	2
DSCP	12	24	26	26	2	2	2
POWE	4	20	21	21	7	8	4
DFPS	5	16	18	22	6	12	0
SIMP	7	18	24	26	2	0	9
ROSE	11	23	26	26	2	0	3
COMP	5	17	24	26	2	0	0
EVOL[†]	17	20	24	28	0	0	0
GRUP[†]	18	22	27	28	0	0	0
REKO[†]	23	24	28	28	0	0	2

Table 6.6 presents again a summary of the number of unconstrained problems that were solved with given accuracy by the search methods under test, together with the number of unsolved problems, the number of cases of fatal execution errors, and the number of cases in which the termination criteria failed.

Constrained Problems

Tables 6.7 and 6.8 show the results of 5 strategies in the 22 constrained problems. Execution errors such as exceeding the number range or dividing by zero did not occur in any case. Neither were there any difficulties in the termination of the searches.

The method of Rosenbrock can only be applied if the starting point of the search lies within the allowed or feasible region. For this reason the initial values of the variables in seven problems had to be altered. All other methods very quickly found a feasible solution to start with. As in the unconstrained problems, the strategies that depend on random numbers were each run three times with different sequences of random numbers. The best of the three results was accepted for evaluation. The results of the complex method and the two membered evolution turned out to be very variable in quality, whereas the multimembered versions of the strategy, especially with recombination, proved to be less influenced by the particular random numbers. Two problems (Problems 2.40 and 2.41) caused great difficulty to all the search methods. These are simple linear programs that can be solved rapidly and exactly by, for example, the simplex method of Dantzig. In

[†] Search terminated twice in each case due to too slow convergence

Table 6.7: Results of all strategies in the second comparison test, constrained problems

Problem No.	ROSE	COMP	EVOL	GRUP	REKO
2.29	3	1	4	3	3
2.30	1	5	1	1	1
2.31	3v	3	1	1	1
2.32	3v	3	1	1	1
2.33	3	2	5	4	1
2.34	1	2	3	3	2
2.35	3v	1	4	4	4
2.36	1	1	1	1	1
2.37	3	1	1	1	1
2.38	3v	3	1	1	1
2.39	3	3	4	4	3
2.40	5	5	5	5	5
2.41	5	5	5	5	5
2.42	3	3	2	2	1
2.43	3v	3	2	2	1
2.44	1	5	1	1	1
2.45	3	2	4	2	1
2.46	3	2	3	3	1
2.47	3v	1	1	1	1
2.48	3v	3	1	1	1
2.49	3	2	4	3	1
2.50	3	1	1	1	1
Sum	62	57	55	50	38

The meaning of the symbols is as in Table 6.5 ; "v" is used in connection with the Rosenbrock method for constrained cases: The starting point had to be displaced since it was not feasible for this method.

each case the closest to the objective was again the $(10\,,100)$ evolution strategy with recombination, but even that result had to be classified as "no solution."

On the whole the evolution methods cope with constrained problems no worse than the Rosenbrock or complex strategies, but they do reveal inadequacies that are not apparent in unconstrained problems. In particular the 1/5 success rule for adapting the variances of the mutation step lengths in the (1+1) evolution strategy appears to be unsuitable for attaining an optimal rate of convergence when several constraints become active.

In problems with active constraints, the tendency of the evolution methods to follow the average gradient trajectory causes the search to come quickly up against one or more boundaries of the feasible region. The subsequent migration towards the objective along

Table 6.8: Summary of the results from Table 6.7

	Total number of problems solved with accuracy class $\delta \leq$				No solution
Strategy	10^{-38}	10^{-8}	10^{-4}	10^{-2}	or $\delta > 10^{-2}$
ROSE	4	4	20	20	2
COMP	6	11	18	18	4
EVOL	10	12	14	19	3
GRUP	10	13	17	20	2
REKO	16	17	19	20	2

such edges takes considerable effort and time. In Figure 6.17 the situation is illustrated for the case of two variables and one constraint.

The contours of the objective function run at a narrow angle α to the boundary of the region. For a mutation to count as successful it must fall within the feasible region as well as improve the objective function value. For simplicity let us assume that all the mutations fall on the circumference of a circle about the current starting point. In the case of many variables this point of view is very reasonable (see Chap. 5, Sect. 5.1). To start with the center of the circle (P_1) will still lie some way from the boundary. If the angle between the contours of the objective function and the edge of the feasible region is small and the step size, or variance of the mutation step size, is large then only a small fraction of the mutations will be successful (thickly drawn part of the circle σ_1). The 1/5 success rule ensures that this fraction is raised to 20%, which if the angle α is small enough can only be achieved by reducing the variance to σ_2. The search point P is driven closer and closer to the boundary and eventually lies on it (P_2). Since there is no longer any finite step size that can provide a sufficiently large success rate, the variance is permanently reduced to the minimum value specified in the program. Depending on the particular problem structure and the chosen values of the parameters in the convergence criteria the search is either slowly continued or it is terminated before reaching the optimum. The more constraints become active during the search, the smaller is the probability that the objective will be closely approached. In fact, even in problems with only two variables and one constraint (Problem 2.46) the angle between the contours and the edge of the feasible region can become vanishingly small in the neighborhood of the minimum.

Similar situations to the one depicted in Figure 6.17 can even arise in unconstrained problems if the objective function displays discontinuities in its first partial derivatives. Examples of this kind of behavior are provided by Problems 2.6 and 2.21. If only a few variables are involved there is still a good chance of reaching the objective. Other search methods, especially those which execute line searches, are generally defeated by such points of discontinuity.

The multimembered evolution strategy, although it works without a rigid step length adaptation, also loses its otherwise reliable convergence characteristics when the region of

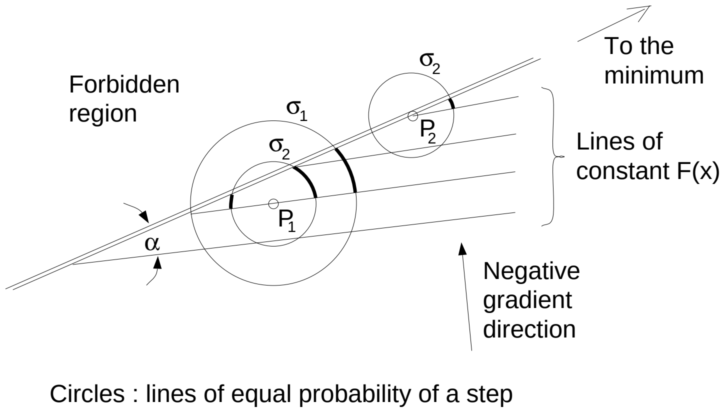
Forbidden region
σ1
σ2
σ2
P1
P2
α
To the minimum
Lines of constant F(x)
Negative gradient direction
Circles : lines of equal probability of a step

Figure 6.17: The situation at active constraints

success is very much narrowed down by constraints. While the individuals are not yet at the edge of the feasible region, those descendants whose step lengths have become smaller have a higher probability of survival. Thus here too the entire population eventually concentrates itself in a smaller and smaller area at the edge of the feasible region.

The theory of the rate of progress in the corridor model did not foresee this kind of difficulty, indeed it gives an optimal success rate, almost the same as in the sphere model, simply because the gradient vector of the objective function always runs parallel to the boundaries. In this case the search weaves backwards and forwards between the center and side of the corridor. The reduced probability of success at multidimensional edges is compensated by the fact that with a uniform probability of occupation over the cross section of the corridor, the space that counts as near to the edges represents a very small fraction of the total. Provided that the success rate is obtained over long enough periods the 1/5 success rule does not lead to permanent reduction of the variances but to a constant near optimal step size (it really fluctuates) that depends only on the width of the corridor and the number of variables.

The situation is happier than in Figure 6.17 if the constraints are given explicitly as

$$x_i \geq a_i \quad \text{or} \quad x_i \leq b_i$$

For any one variable, the region of success at a boundary is reduced by one half. If at some position m variables are each bounded on one side, then on average it costs 2^m mutations before one lands within the feasible region. Here again, the 1/5 success rule for $m > 2$ will continuously reduce the variances until they reach their minimum value. Depending on the route chosen by the search process the limiting values of the variances, which are individually adjustable for each variable, will be reached at different times. Their relative values thereby alter, and with the new combination of step lengths the convergence can be faster.

The extra flexibility of the multimembered evolution strategy with recombination, in which the variances of the changes in the variables are individually adaptable during the whole of the optimization process, is a very clear advantage in solving constrained problems. Suitable combinations of variances are set up in this case before the smallest possible step lengths are reached. Thus the total computation time is reduced and the final accuracy is better. The recombination option also appears to have a beneficial effect at boundaries that are not explicit; it clearly improves the chance that descendants, even with a larger step size, will be successful near the boundary. In any case the population clusters more slowly together than when there is no recombination.

Global Convergence Properties

Among the 50 test problems there are 8 having at least a second local minimum besides the global one. In the reliability test, the accuracy achieved was only assessed with respect to the particular optimum that was being approximated. What now is the capability of each strategy for locating global minima? Several problems were specifically designed to investigate this question by having very many local optima, namely Problems 2.3, 2.26, 2.30, and 2.44. In Table 6.9 this aspect of the test results is evaluated.

Except for one problem (Problem 2.32), whose global minimum was found by all the strategies under test, the method of Rosenbrock only converged to local optima. The complex method and the (1+1) evolution strategy were only better in one case: namely, in Problem 2.45 they both approached the global minimum.

Table 6.9: Results of all strategies in the second comparison test: global convergence properties

Problem	F	G	L	H	D	D	P	D	S	R	C	E	G	R
	I	O	A	O	S	S	O	F	I	O	O	V	R	E
	B	L	G	J	C	C	W	P	M	S	M	O	U	K
No.	O	D	R	E	G	P	E	S	P	E	P	L	P	O
2.3	L1	L1	L3	L1	L7	L7	L1	L3	L1	L6	L1	Lm	G	G
2.36	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	G	G
2.30										L4	L1	Lm	G	G
2.32										G	G	G	G	G
2.44										L1	L1	L1	G	G
2.45										L	G	G	G	G
2.47										L3	L1	L2	G	G
2.48										L2	Lm	Lm	GL	GL

Meaning of symbols:

L Search converges to local minimum.
L3 Search converges to the 3rd local minimum (in order of decreasing objective function values).
Lm Search converges to various local minima depending on the random numbers.
G Search converges to global minimum.
GL Search converges to local or global minimum depending on the random numbers.

The multimembered evolution strategy displays much better global convergence properties, with or without recombination. Although its actual path through the space was determined by chance, it always found its way to the absolute minimum. Only in Problem 2.48 was the global optimum not always approached. In this case the feasible region is not simply connected: Between the starting point and the global minimum there is no connecting line that does not pass through the region excluded by constraints. The path of the simple evolution strategy and the initial condition of the complex method are also both dependent on the particular sequence of pseudorandom numbers; however, the main difference between the results of three trials in each case was simply that different local minima were approached. In one case the (1+1) evolution rejected 33 local optima only to converge at the 34th (Problem 2.3).

In spite of the good convergence properties of the multimembered evolution manifested in the tests, a certain measure of scepticism is called for. If the search is started with only small step lengths in the neighborhood of a local minimum, while the global minimum is far removed and is surrounded by only a relatively small region with small objective function values, then the probability of getting there can be very small.

If in addition there are very many variables, so that the step sizes of the mutations are small compared to the Euclidean distance between two points in $\mathbb{R}^n$, the search for a global optimum among many local optima is like the proverbial search for a needle in a haystack. Locating singular minima, even with only a few variables, is a practically hopeless task. Although the multimembered evolution increases the probability of finding global minima compared to other methods, it cannot guarantee to do so because of its basically sequential character.

6.3.3.3 Third Test: Non-Quadratic Problems with Many Variables

In the first series of tests we investigated the rates of convergence for a quadratic objective function, and in the second the reliability of convergence for the general non-linear case. The aim of the third test is now to study the computational effort required for non-quadratic problems. Because of their small number of variables, the problems of the second test series appear unsuitable for this purpose, as rates of convergence and computation times are only of interest in relation to the number of variables. The construction of non-quadratic objective functions of a type that can also be extended to an arbitrary number of variables is not a trivial problem. Another reason, however, for this third strategy comparison being restricted to only 10 different problems is that it required a large amount of computation time. In some cases CPU times of several hours were needed to test just one strategy on one problem with a particular number of variables. Seven of the problems are unconstrained and three have constraints. Appendix A, Section A.3 contains the mathematical formulation of the problems together with their solutions.

The procedure followed was the same as in the first test. Besides the termination criterion specific to each strategy, which demanded maximum accuracy, a further convergence criterion was applied in common to all strategies. According to the latter the search was to be ended when a specified distance had been covered from the starting point towards the minimum. The number of variables was varied up to the maximum allowed by the storage capacity, taking the values 3, 10, 30, 100, 300, and 1000. Of course, if a problem with, for example, 30 variables could not be solved by a strategy, or if no result was forthcoming at the end of the maximum computation time of 8 hours, the number of variables was not increased any further.

As in the first test, the initial conditions were specified by

$$x_i^{(0)} = x_i^* + \frac{(-1)^i}{\sqrt{n}} \ , \qquad i \, = \, 1(1)n$$

Two exceptions are Problem 3.3 with

$$x_i^{(0)} = x_i^* + \frac{\pi \, (-1)^i}{10 \, \sqrt{n}}$$

to ensure that the search always converged to the desired minimum and not to one of the many others of equal value; and Problem 3.10 with

$$x_i^{(0)} = x_i^* + \frac{1}{\sqrt{n}}$$

to start the search within the feasible region. Problems 3.8 and 3.9, whose minima are at infinity, required special treatment of the starting point and termination conditions (see Appendix A, Sect. A.3).

The results are presented in Table 6.10. For comparison, some of the results of the first test (Problem 1.1) are also displayed. The numbers enable one to assess critically on the one hand the reliability of a strategy and on the other the computation times it requires.

Table 6.10: Results of all strategies in the third comparison test

The following notation is used in the tables:

n: Number of variables

Case: A label for the convergence behavior, taking the values:

1 Normal end of search; required approximation to the objective was achieved.

2 The search was ended before reaching the desired accuracy.

3 The search became unending without converging; it had to be terminated externally.

4 The maximum computation time of 8 hours was insufficient to end the search successfully (occasionally more computation time was invested in trials with the multimembered evolution strategy that promised to be successful).

- No trial was undertaken.

1(2) Depending on the sequence of random numbers various cases occurred; the entries in the table refer to the first case defined.

OFC: Number of objective function calls.

CFC: Number of constraint function calls.

Time: Computation time in seconds (CPU time).

Iterations, cycles, exploratory cycles, line searches, orthogonalizations, restarts, etc., were counted as in the first comparison test.
Fatal execution errors were only registered in the Powell and DFPS methods and it is not further specified here in which problems they occurred. As a rule the same types of problem were involved as in the second test.
In unconstrained problems no numbers are tabulated for the number of objective function calls made by the evolution strategies. This can be calculated from the number of mutations or generations as follows:

EVOL: 1 + number of mutations

GRUP, REKO: 10 + 100 times number of generations

(continued)

Table 6.10 continued : Coordinate strategies FIBO, GOLD, LAGR (from top to bottom)

Probl.	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time
	n = 3				n = 10				n = 30				n = 100				n = 300				n = 1000			
1.1	1	1	158	0.13	1	1	456	0.53	1	1	1242	3.07	1	1	3870	26.5	1	1	10562	210	1	1	38701	2500
3.1	1	3	415	0.38	1	5	1997	4.12	1	4	4270	18.90	1	2	7041	93.5	1	1	11244	434	1	1	35300	4590
3.2	1	10	1250	1.04	1	9	3470	5.90	1	6	6472	27.50	1	2	7139	94.5	1	1	11218	435	1	1	35303	4730
3.3	1	4	630	4.26	2	18	4381	286.00	2				-				-				-			
3.4	1	1	192	0.14	1	1	567	0.72	1	1	1709	4.68	1	1	5193	40.0	1	1	15127	334	1	1	49255	3670
3.5	2	1	85	0.08	2				2				2				2				2			
3.6	1	1	192	0.18	1	1	567	1.10	1	1	1709	6.92	1	1	5193	53.9	1	1	15127	428	1	1	49255	4590
3.7	1	2	183	0.18	1	1	296	0.48	1	1	816	2.92	1	2	4818	52.3	1	2	14634	478	1	2	47799	6770
1.1	1	1	158	0.10	1	1	458	0.51	1	1	1242	3.14	1	1	3870	27.6	1	1	10562	221	1	1	38703	2670
3.1	1	3	415	0.34	1	5	1993	3.38	1	4	4441	18.70	1	2	7017	101.0	1	1	11244	430	1	1	35300	4740
3.2	1	10	1250	0.84	1	9	3543	5.54	1	6	6347	28.60	1	2	7013	91.2	1	1	11218	439	1	1	35303	4580
3.3	1	4	630	3.96	2	9	2143	136.00	2				-				-				-			
3.4	1	1	192	0.12	1	1	589	0.68	1	1	1715	4.68	1	1	5207	42.4	1	1	15111	332	1	1	49132	3680
3.5	2	1	85	0.08	2				2				2				2				2			
3.6	1	1	192	0.14	1	1	589	1.10	1	1	1715	6.98	1	1	5207	53.1	1	1	15111	440	1	1	49132	4750
3.7	1	2	183	0.18	1	1	296	0.48	1	1	816	3.00	1	2	4849	52.5	1	2	14633	476	1	2	47767	6720
1.1	1	1	85	0.04	1	1	271	0.30	1	1	781	1.80	1	1	2501	17.3	1	1	7201	153	1	1	24001	1620
3.1	1	2	198	0.16	1	5	1285	2.22	1	4	2816	12.00	1	2	4808	66.3	1	1	8073	345	1	1	26415	3600
3.2	1	10	745	0.56	1	9	2142	3.52	1	6	4012	17.60	1	2	4790	64.4	1	1	8046	345	1	1	26332	3500
3.3	1	4	321	2.08	2	24	3178	199.00	2				-				-				-			
3.4	1	1	140	0.10	1	1	436	0.56	1	1	1274	3.20	1	1	3695	30.1	1	1	11270	291	1	1	35247	2600
3.5	2	1	82	0.06	2				2				2				2				2			
3.6	1	1	138	0.12	1	1	437	0.64	1	1	1274	4.62	1	1	3695	35.3	1	1	11270	319	1	1	35247	3240
3.7	1	1	87	0.06	1	1	264	0.42	1	1	785	2.80	1	1	2509	25.5	1	1	7204	240	1	2	42993	7890

Table 6.10 continued : HOJE - Direct search of Hooke and Jeeves

Probl.	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time	case	cycles	OFC	time
	n = 3				n = 10				n = 30				n = 100				n = 300				n = 1000			
1.1	1	4	20	0.02	1	3	48	0.06	1	3	168	0.43	1	2	352	2.37	1	9	4954	100	1	12	23505	1460
3.1	1	4	20	0.02	1	3	48	0.08	1	3	168	0.74	1	2	352	4.82	1	33	18612	784	1	21	42004	5710
3.2	1	4	20	0.02	1	3	48	0.10	1	3	168	0.74	1	2	352	4.78	1	32	17714	758	1	21	42004	5440
3.3	1	3	19	0.12	1	12	237	16.70	1	130	7493	4210.00	-				-				-			
3.4	1	4	20	0.02	1	3	48	0.06	1	3	168	0.48	1	2	352	2.86	1	9	4954	119	1	12	23505	1700
3.5	2	25	151	0.18	2				2				2				2				2			
3.6	1	4	20	0.02	1	3	48	0.08	1	3	168	0.70	1	2	352	3.72	1	9	4954	153	1	12	23505	2160
3.7	1	4	20	0.02	1	3	48	0.08	1	3	168	0.62	1	2	352	3.68	1	2	4954	137	1	12	23505	2410

ROSE - Rosenbrock method with Gram-Schmidt orthogonalization

Probl.	case	orth.	OFC	CFC	time	case	orth.	OFC	CFC	time	case	orth.	OFC	CFC	time	case	orth.	OFC	CFC	time
	n = 3					n = 10					n = 30					n = 75 (max)				
1.1	1	1	27		0.08	1	2	120		0.91	1	0	121		1.18	1	2	899		145
3.1	1	3	41		0.14	1	3	161		1.36	1	3	575		18.70	1	3	2352		242
3.2	1	5	98		0.30	1	4	282		2.10	1	3	575		19.10	1	4	3879		342
3.3	1	3	45		0.40	1	1	73		5.02	1	8	3077		1690.00	-				
3.4	1	2	44		0.14	1	4	279		1.98	1	0	121		1.20	1	47	83059		4660
3.5	1	2	29		0.12	1	3	295		2.02	1	0	121		1.36	1	9	7337		728
3.6	1	2	44		0.14	1	4	279		2.06	1	0	121		1.32	1	47	83059		4830
3.7	1	2	29		0.10	1	2	152		1.10	1	0	121		1.32	1	3	1871		236
3.8	1	1	28	101	0.16	1	1	91	1128	1.60	1	1	241	7861	17.30	1	0	226	33448	85
3.9	1	1	28	28	0.10	1	6	427	309	3.46	1	1	512	215	9.76	1	2	2833	969	194
3.10	1	8	268	367	1.16	2	12	2953	9766	30.50	2					-				

Table 6.10 continued : DSCG - Davies-Swann-Campey method with Gram-Schmidt orthonormalization

Probl.	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time
	n = 3					n = 10					n = 30					n = 75 (max)				
1.1	1	0	3	20	0.04	1	0	10	56	0.20	1	0	30	136	1.18	1	0	75	338	6.10
3.1	1	1	6	32	0.08	1	2	30	119	0.86	1	1	91	321	7.58	1	1	150	563	78.40
3.2	1	3	12	48	0.12	1	3	40	147	1.12	1	3	151	535	18.70	1	1	150	563	79.00
3.3	1	2	9	35	0.28	1	8	112	377	26.00	2	28	1087	3366	2030.00	-				
3.4	1	0	3	20	0.06	1	0	10	56	0.20	1	0	30	136	1.18	1	0	75	338	6.68
3.5	1	1	6	30	0.08	1	0	10	47	0.20	1	0	30	165	1.58	1	1	150	490	78.20
3.6	1	0	3	20	0.06	1	0	10	56	0.22	1	0	30	136	1.48	1	0	75	338	7.08
3.7	1	1	6	30	0.10	1	0	10	56	0.22	1	0	30	136	1.26	1	1	150	564	76.90

DSCP - Davies-Swann-Campey method with Palmer orthonormalization

Probl.	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time	case	orth.	lin. search	OFC	time
	n = 3					n = 10					n = 30					n = 75 (max)					n = 95 (max)				
1.1	1	0	3	20	0.04	1	0	10	56	0.22	1	0	30	136	1.16	1	0	75	338	6.10	1	0	95	448	9.49
3.1	1	1	6	32	0.08	1	2	30	119	0.58	1	1	91	321	3.58	1	1	150	563	14.90	1	0	95	428	11.90
3.2	1	3	12	48	0.12	1	3	40	147	0.72	1	3	151	527	6.56	1	1	150	563	14.80	1	1	190	713	23.20
3.3	1	2	9	35	0.28	1	8	112	383	25.50	1	28	932	2924	1670.00	-					-				
3.4	1	0	3	20	0.06	1	0	10	56	0.20	1	0	30	136	1.28	1	0	75	338	6.36	1	0	95	428	9.96
3.5	1	1	6	31	0.08	1	0	10	47	0.20	1	0	30	165	1.60	1	1	150	490	12.90	1	0	95	334	9.22
3.6	1	0	3	20	0.06	1	0	10	56	0.22	1	0	30	136	1.36	1	0	75	338	7.18	1	0	95	428	11.1
3.7	1	1	6	30	0.08	1	0	10	56	0.22	1	0	30	136	1.26	1	1	150	564	13.70	1	0	95	428	11.2

Table 6.10 continued : POWE - Powell's method of conjugate directions

Probl.	case	iterations	lin. search	OFC	time	case	iterations	lin. search	OFC	time	case	iterations	lin. search	OFC	time	case	iterations	lin. search	OFC	time	case	iterations	lin. search	OFC	time
	n = 3					n = 10					n = 30					n = 100					n = 135 (max)				
1.1	1	1	3	11	0.02	1	1	10	32	0.12	1	1	30	92	0.60	1	1	100	202	3.72	1	1	135	407	8.60
3.1	1	2	7	32	0.08	1	2	21	84	0.36	1	3	91	304	2.50	1	2	200	701	15.10	1	1	135	541	15.50
3.2	1	3	11	38	0.12	1	3	32	112	0.46	1	3	91	288	2.32	1	2	200	701	15.00	1	1	135	541	15.50
3.3	1	2	7	23	0.18	1	9	96	257	16.90	1	23	700	1963	1100.00	3					-				
3.4	1	1	3	20	0.06	1	2	20	128	0.44	1	1	30	152	0.94	1	1	100	502	7.82	1	1	135	811	16.20
3.5	2	1	3	55	0.14	2					2					2					2				
3.6	1	1	3	20	0.06	1	2	20	128	0.48	1	1	30	152	1.14	1	1	100	502	8.74	1	1	135	811	19.00
3.7	1	4	13	166	0.46	3					3					3					3				

DFPS - Stewart's modification of the Davidon-Fletcher-Powell method

Probl.	case	iterations	OFC	time	case	iterations	OFC	time	case	iterations	OFC	time	case	iterations	OFC	time	case	iterations	OFC	time
	n = 3				n = 10				n = 30				n = 100				n = 180 (max)			
1.1	1	1	10	0.02	1	1	24	0.06	1	1	64	0.32	1	1	204	3.19	1	1	364	9.56
3.1	1	3	20	0.06	1	3	48	0.20	1	4	160	1.82	1	5	612	22.00	1	6	1274	84.40
3.2	1	4	25	0.06	1	4	65	0.30	1	4	160	1.78	1	5	612	23.10	1	6	1274	84.20
3.3	1	2	15	0.12	1	4	61	3.96	1	45	1640	932.00	-				-			
3.4	1	1	10	0.02	1	1	24	0.06	1	1	64	0.34	1	1	204	3.34	1	1	364	10.20
3.5	2	1	11	0.02	2				2				2				2			
3.6	1	1	10	0.02	1	1	24	0.08	1	1	64	0.40	1	1	204	3.92	1	1	364	11.80
3.7	1	15	101	0.26	1	16	307	1.42	2	15	780	7.84	2				2			

Table 6.10 continued : SIMP - Simplex method of Nelder and Mead (with restart rule)

Probl.	case	restarts	OFC	time	case	restarts	OFC	time	case	restarts	OFC	time	case	restarts	OFC	time	case	restarts	OFC	time
	n = 3				n = 10				n = 30				n = 100				n = 135 (max)			
1.1	1	0	28	0.09	1	0	138	1.49	1	0	664	37.4	1	0	1789	862	1	1	5142	5270
3.1	1	0	34	0.10	1	0	152	1.56	1	0	1514	95.2	1	1	42099	25800	4			
3.2	1	0	40	0.12	1	0	300	3.54	1	0	1277	79.7	1	0	10082	6190	4			
3.3	1	0	32	0.28	1	0	163	12.10	1	1	12936	8130.0	-				-			
3.4	1	0	25	0.08	1	0	163	1.62	1	7	17357	1100.0	4				-			
3.5	1	0	32	0.08	1	185	22362	206.00	2	547	101142	4030.0	-				-			
3.6	1	0	25	0.08	1	0	163	1.68	1	7	17357	1140.0	4				-			
3.7	1	0	28	0.08	1	48	3968	39.10	1	181	21387	1020.0	4				-			

COMP - Complex method of Box (no. of vertices = 2n)

Probl.	case	OFC	CFC	time	case	OFC	CFC	time	case	OFC	CFC	time	case	OFC	time
	n = 3				n = 10				n = 30				n = 95 (max)		
1.1	1	69		0.22	1	535		6.72	1	2621		211	1	14902	11000
3.1	1	76		0.28	1	691		9.30	1	2770		247	1	17397	13100
3.2	1	83		0.28	1	527		6.99	1	3092		259	1	17431	13300
3.3	1	58		0.55	1	966		74.30	2(4)	19407		12100	-		
3.4	1	74		0.24	1(2)	529		6.71	1(2)	4722		402	4		
3.5	1	57		0.23	1	1266		17.20	2(4)	16936		1450	4		
3.6	1	85		0.32	1	556		7.34	1(2)	9011		781	4		
3.7	1	76		0.30	1	587		7.68	1	9537		840	1	32196	25300
3.8	1	40	186	0.33	1	207	4680	8.32	1	1041	75867	240	4		
3.9	1	35	44	0.19	1	384	600	8.46	2	2622	4508	357	-		
3.10	1	33	208	0.25	1(4)	486	8919	12.10	4				-		

Table 6.10 continued : EVOL - (1+1) evolution strategy

Probl.	case	mutations	OFC	CFC	time	case	mutations	OFC	CFC	time	case	mutations	OFC	CFC	time
	n = 3					n = 10					n = 30				
1.1	1	49			0.17	1	224			1.74	1	630			14.0
3.1	1	67			0.22	1	221			1.88	1	730			16.6
3.2	1	179			0.52	1	537			4.46	1	1041			26.5
3.3	1	61			0.50	1	2244			156.00	1	7103			4060.0
3.4	1	66			0.20	1	257			2.18	1	10939			244.0
3.5	1	74			0.36	2	200			1.82	2				
3.6	1	89			0.32	1	325			2.87	1	15769			365.0
3.7	1	63			0.20	1	206			1.74	1	1154			26.7
3.8	1	99	45	301	0.54	1	319	136	3980	5.80	1	668	300	24629	46.2
3.9	1	78	31	79	0.29	1	388	148	389	3.33	1	1435	579	1436	34.5
3.10	1	925	592	3913	4.06	1	59757	39076	802511	824.00	4				

Probl.	case	mutations	OFC	CFC	time	case	mutations	OFC	CFC	time	case	mutations	OFC	CFC	time
	n = 100					n = 300					n = 1000				
1.1	1	2192			149	1	6666			1310	1	23607			15600
3.1	1	2185			164	1	7638			1700	1	23374			17100
3.2	1	2208			188	1	6916			1480	1	23819			17200
3.3	-					-					-				
3.4	1	47720			3320	4					-				
3.5	2					2					2				
3.6	1	37985			2710	4					-				
3.7	1	5145			389	1	14818			3060	4				
3.8	1	1696	998	255392	803	1	3161	3129	1883309	14100	-				
3.9	1	4633	1870	4634	336	1	13035	5374	13036	2830	-				
3.10	-					-					-				

Table 6.10 continued : GRUP - (10,100) evolution strategy

Probl.	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time
	n = 3					n = 10					n = 30				
1.1	1	4			1.81	1	17			16.8	1	55			145
3.1	1	7			3.18	1	20			20.3	1	77			201
3.2	1	4			1.56	1	25			24.1	1	67			194
3.3	1	5			5.02	1	23			164.0	1	1462			82100
3.4	1	3			1.44	1	18			20.9	1(2)	848			2140
3.5	1	4			1.90	1	21			21.8	2	481			1300
3.6	1	5			2.26	1	19			19.6	1	446			1130
3.7	1	5			2.08	1	20			20.8	1	80			200
3.8	1	9	540	2940	5.19	1	25	1355	33663	48.0	1	51	2983	226805	390
3.9	1	6	414	610	2.68	1	37	2067	3710	35.4	1	96	4801	9610	239
3.10	1	103	6937	45174	58.0	1	3707	230227	4745318	5360	4				

Probl.	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time
	n = 100					n = 300					n = 435 (max)				
1.1	1	194			1600	1	551			13600	1	854			28300
3.1	1	213			1730	1	677			16400	1	865			30400
3.2	1	199			1660	1	604			14600	1	940			33100
3.3	-					-					-				
3.4	2	1653			13200	-					-				
3.5	-					-					-				
3.6	2	1650			13200	-					-				
3.7	1	473			3790	1	1655			38100	-				
3.8	1	109	5925	1614891	5090	4									
3.9	1	329	15941	32915	2620	1	986	47168	98673	23200	-				
3.10	-					-					-				

Table 6.10 continued : REKO - (10,100) evolution strategy with recombination

Probl.	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time
	n = 3					n = 10					n = 30				
1.1	1	4			2.67	1	13			23.3	1	34			177
3.1	1	5			3.56	1	15			31.3	1	32			170
3.2	1	6			4.32	1	11			19.8	1	35			198
3.3	1	1			1.40	1	14			122.0	1	1365			79700
3.4	1	3			2.06	1	15			26.3	1	28			152
3.5	1	5			3.43	1	19			35.7	1(2)	213			1120
3.6	1	3			2.22	1	12			21.5	1	29			148
3.7	1	5			3.60	1	17			30.4	1	41			209
3.8	1	10	584	3549	8.25	1	20	1410	30006	60.5	1	45	3920	242482	526
3.9	1	10	638	1010	7.09	1	28	1978	2810	53.9	1	92	6321	9210	487
3.10	1	8	462	3321	6.88	1	456	27875	586357	1050.0	4				

Probl.	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time	case	generations	OFC	CFC	time
	n = 100					n = 300					n = 435 (max)				
1.1	1	84			1420	1	180			9340	1	289			21100
3.1	1	80			1420	1	206			10800	1	305			22700
3.2	1	77			1350	1	212			10800	1	257			19400
3.3	-					-					-				
3.4	1	77			1250	1	210			10800	1	305			21900
3.5	2	1653			28200	-					-				
3.6	1	82			1420	1	205			10600	1	307			23000
3.7	1	137			2370	1	399			20800	-				
3.8	1	128	12354	2529790	9080	4					-				
3.9	1	186	13547	18610	3110	1	494	36708	49410	25200	-				
3.10	-					-					-				

With only three variables, nearly all the problems were solved perfectly by all strategies; i.e., the required approximation to the objective was achieved. The only exception is Problem 3.5, which ended in failure for the coordinate strategies, the method of Hooke and Jeeves, and the methods of Powell and of Davidon-Fletcher-Powell-Stewart. In apparent contradiction to this, the corresponding Problem 2.21 for $n = 5$ was satisfactorily solved by the Hooke-Jeeves strategy and the DFPS method. The causes are to be found in the different initial values of the variables. With the variable metric method, fatal execution errors occurred in both cases.

If there are 10 or more variables, even the two membered evolution strategy does not find the minimum in Problem 3.5, due to the extremely unfavorable starting point. The probability of making from there a first step with a lower objective function value is 2^{-n}. Thus with many variables, the termination condition is usually met before a single success has been scored. The simplex method of Nelder and Mead with $n = 10$ took 185 restarts to reach the desired approximation to the objective. For more than 10 parameters the solution can no longer be sufficiently well approximated in spite of an increasing number of restarts. With stricter accuracy requirements the simplex method fails much sooner (Problem 2.21 with $n = 5$).

The complex strategy likewise was no longer able to solve the same problem for $n \geq 30$. Depending on the sequence of random numbers it either ended the search before achieving the required accuracy, or it was still far from the minimum when the allowed computation time (8 hours) expired. The multimembered evolution strategy also proved to be dependent, although less strongly, on the particular sequence of random numbers. The version without recombination failed on Problem 3.5 for $n \geq 30$; with recombination it failed for $n \geq 100$. Without recombination and for $n \geq 100$ it ended the minimum search prematurely also in Problems 3.4 and 3.6. The simplex and complex methods had convergence difficulties with both types of objective function, usually even for only a few variables. Several times they had to be interrupted because of exceeding the time limit. Further details can be found in the tables and Appendix A, Section A.3.

The search for the minima in Problems 3.4 and 3.6 presents no difficulties to the coordinate strategies, and the methods of Hooke and Jeeves, Rosenbrock, Davies-Swann-Campey, Powell, and Davidon-Fletcher-Powell-Stewart. The three rotating coordinate strategies are the only ones that manage to solve Problem 3.5 satisfactorily for any number of variables. Nevertheless it would be hasty to conclude that these methods are therefore clearly better than the others; an attempt to analyze the reasons for their success reveals that only slight changes in the objective functions are enough to undermine their apparently advantageous way of working.

The significant difference in this respect between the above group of strategies and the others (complex, simplex, and evolution strategies) is that the former operate with a much more limited set of search directions than the latter. There are usually only n directions, e.g., the n coordinate directions of the axes-parallel search methods, compared to an infinite number (in principle) in the evolution methods. In the case of Problems 3.4 to 3.6 the most favorable search directions are the n directions of the unit vectors. All methods with one dimensional minimizations use precisely these directions in their first iteration cycle, so they do not usually require any further iterations to achieve the required accuracy. By keeping the starting conditions the same but rotating the coordinates with respect to the contours of the objective function (Problem 3.6), or slightly tilting the contours with respect to the coordinate axes (Problem 3.5), or both together (Problem 3.4), one could easily cause all the line searches to fail. On the other hand the strategies without line searches would not be impaired by these changes. Thus the advantage of selected directions can turn into a disadvantage. These coordinated strategies can never solve the problem referred to, whereas, as we have seen, the strategies that have a large set of search directions at their disposal only fail when a particular number of variables is exceeded. Problems 3.4 and 3.6 are therefore suitable for assessing the reliability of simplex, complex, and evolution strategies, but not for the other methods. Together they belong to the type of problems which Himmelblau designates as "pathological."

Leaning more to the conservative side are the several times continuously differentiable objective functions of Problems 3.1, 3.2, 3.3, and 3.7. The first two problems were tackled successfully by all the strategies for any number of variables. The simplex method did, however, need at least one restart for Problem 3.1 with $n \geq 100$. For 135 variables it exceeded the time limit before Problems 3.1 and 3.2 were solved to sufficient accuracy.

Problem 3.3 gave trouble to several search procedures when there were 10 or more variables. The coordinate strategies were the first to fail. For only $n = 10$, the step lengths of the line searches would have had to be smaller than allowed by the number precision of the computer used. At $n = 30$, the DSC strategy with Gram-Schmidt orthogonalization also ends without having located the minimum accurately enough. The simplex method with one restart still found the solution for $n = 30$, but the complex strategy failed here, either by premature termination of the search or by reaching the maximum permitted computation time. Problem 3.3, because the cost per objective function evaluation increases as $O(n^2)$, requires the longest computation times for its solution. Since the objective function also took $O(n^2)$ units of storage, this problem could not be used for more than 30 variables.

Problem 3.7, like the analogous Problem 2.31, gave trouble to the two quadratically convergent strategies. The method of Powell was only successful for $n = 3$. For more variables it became stuck in the search process without the termination rule taking effect. The variable metric strategy behaved in just the same way. For $n \geq 30$, it no longer came as near as required to the optimum. Under the stricter conditions of the second set of tests it failed already at $n = 5$. With both methods fatal execution errors occurred during the search. No other direct search strategies had any difficulty with Problem 3.7, which is a simple 10th order polynomial. Only the simplex method would not have found the solution sufficiently accurately without the restart rule. For $n = 100$, it reached the time limit before the search simplex had collapsed for the first time.

The advantage shown by the complex strategy was due to the complex's having $2n$ vertices, which is almost twice as many as the $n + 1$ of the simplex. An attempt to solve Problems 3.1 to 3.10 for $n = 30$ with a complex constructed of 40 points failed completely. The search ended, in every case, without having reached the required accuracy.

How do the computation times compare when the problems are no longer only quadratically non-linear? For solving the "pathological" Problems 3.4 to 3.6 all the methods with a line search take about the same times, with the same number of variables, as they do for solving the simple quadratic Problem 1.1, if indeed they actually can find a solution. With any of the remaining methods the computation times increase somewhat more steeply with the number of variables, up to the limiting number beyond which convergence cannot be guaranteed in every case.

The solution times for Problems 3.1 and 3.2 usually turn out to be several times greater than those for Problem 1.1. The cost of the coordinate strategies is up to 1000% more for a few variables, which reduces to 100% as the number of variables increases. As in the case of Problem 1.1, the solution times for Problems 3.1 and 3.2 using the method of Hooke and Jeeves increase somewhat faster than the square of the number of variables. For very many variables 250% more computation time is required.

For $n \geq 30$, the Rosenbrock method requires 70% to 250% more time (depending on the number of orthogonalizations) for the first two problems of the third set of tests than for the simple quadratic problem. The computation time still increases as $O(n^3)$ in all cases because of the costly procedure of rotating the coordinates. For example, for $n = 75$, up to 90% of the total time is taken up by the orthogonalizations. The DSC strategies reached the desired accuracy in Problem 1.1 without orthogonalizations. Since solving Problems 3.1 and 3.2 requires more than n line searches in each case, the computation times differ significantly, depending on the chosen method of orthogonalization. Palmer's program holds down the increase in the computation times to $O(n^2)$ whereas the Gram-Schmidt method leads to an $O(n^3)$ increase. It therefore is not meaningful to quote the extra cost as a percentage with respect to Problem 1.1. In the extreme case instead of 6 seconds at $n = 75$ the procedure took nearly 80 seconds.

The method of Powell requires two to four times as much time, depending on whether one or two extra iterations are needed. However, even for the same number of iterations, i.e., also with the same number of line searches ($n = 135$), the number of function calls in Problems 3.1 and 3.2 is greater than in Problem 1.1. The reason for this is that in the quadratic reference problem a simplified form of the parabolic interpolation can be used. The variable metric strategy, in order to solve the two non-quadratic problems (Problems 3.1 and 3.2) with $n = 180$, requires about nine times as much computation time as for Problem 1.1. This factor increases with n since the number of gradient determinations increases gradually with n.

The pattern of behavior of the simplex method of Nelder and Mead is very irregular. If the number of variables is small, the computation times for all three problems are about equal. However, for $n = 100$, Problem 3.2 requires about seven times as much time to be solved as Problem 1.1; and, because of a restart, Problem 3.1 requires even thirty times as much. With $n = 135$, neither of the two non-quadratic problems can be solved within 8 hours, whereas 1.5 hours are sufficient for Problem 1.1. On the other hand the complex strategy requires only slightly more time, about 20%, than in the simple quadratic case, provided $2n$ vertices are taken. The time taken by this method on the whole for all problems, however, exhibits the strongest rate and range of variation with the number of parameters.

The evolution strategies prove to be completely unaffected by the altered topology of the objective function as compared with the case of spherically symmetrical contour surfaces. Within the expected deviations, due to different sequences of random numbers, the measured computation times for all three problems are equal. The results show that Rechenberg's (1973) theory of the rate of progress, which does not assume a quadratic objective function but merely concentric hypersphere contour surfaces, is valid over a wide range of conditions. Even more surprising, however, is the behavior of the (10 , 100) evolution method with recombination in the solution of Problems 3.4 and 3.6, whose objective functions have discontinuous first derivatives, i.e., their contour surfaces display sharp edges and corners. The mixing of the components of variables representing individuals on different sides of a discontinuity appears sometimes to have a kind of smoothing effect. In any case it can be seen that the strategy with recombination needs no more computation time or objective function calls for Problems 3.4 and 3.6 than for Problems 1.1, 3.1, and 3.2.

With all the methods under test, the computation times for solving Problem 3.7 are about twice as high as those measured in the simple quadratic case. Only the simplex method is significantly more demanding of time. Since the search simplex frequently collapses in on itself it must repeatedly be reinitialized.

Since Problem 3.3 could only be tackled with 3, 10, and 30 variables it is not easy to analyze the resulting data. In addition, the dependence of the increase in difficulty on the number of parameters is not so clear-cut in this problem. Nevertheless the results seem to indicate that at least the number of objective function calls, in many strategies, increases with n in a way similar to that in the pure quadratic Problem 1.2. Because an objective function evaluation takes about $O(n^2)$ operations in Problem 3.3, the total cost generally increases as one higher power of n than in Problem 1.2. The cost of the variable metric strategy and both versions of the (10 , 100) evolution strategy seems to increase even more rapidly. In the latter case there is a suspicion that the chosen initial step lengths are too large for this problem when there are very many variables. Their reduction to a suitable size then takes a few additional generations. The two membered evolution strategy, which is able to adjust unsuitable initial step lengths relatively quickly, needed about the same number of mutations for both Problems 1.2 and 3.3. Since only one experiment per strategy and number of variables was performed, the effect of the particular sequence of random numbers on the recorded computation times is not known. The particularly advantageous behavior of the DFPS method on exactly quadratic objective functions is clearly wasted once the problem deviates from this model structure; in fact it seems that the search process is appreciably held back by an interpretation of the measured data in terms of an inappropriate internal model.

So far we have only discussed the results for the seven unconstrained problems, since they were amenable to solution by all the search strategies. Problem 3.8, with constraints, corresponds to the second model function (corridor model) for which Rechenberg (1973) has obtained theoretically the rate of progress of the two membered evolution strategy with optimal adaptation of variances. According to his analysis, one expects a linear rate of convergence increasing with the width of the corridor and inversely proportional to the number of variables. The results of the third set of tests confirm that the number of mutations or generations increases linearly with n if the width of the corridor and the reference distance to be covered are held constant. The picture for the Rosenbrock strategy is as usual: the time consumption increases as $O(n^3)$ again. The point at $n = 75$

departs from the general trend of the others simply because no orthogonalizations were performed in this case. But the difference is not dramatic, because the cost of testing the constraints is of the same order of magnitude as that of rotating the coordinates. The complex method takes computation times that initially increase somewhat more rapidly than $O(n^3)$. This corresponds to a greater than linearly increasing number of objective function evaluations. As we have already seen in other problems, the increase becomes even steeper as the number of parameters increases. With $n = 95$ variables, the required distance was only partially covered within the maximum computation time.

Problem 3.9 represents a modification of Problem 3.8 with respect to the constraints. In place of the $(2n - 2)$ linear constraints, the corridor is bounded by a single non-linear boundary condition. The cost of testing the feasibility of an iteration point is thereby greatly reduced. The number of mutations or generations of the evolution strategies is higher than in Problem 3.8 but still increases as $O(n)$; the computation times in contrast to Problem 3.8 only increase as $O(n^2)$. The Rosenbrock method also has no difficulty with this problem, although the necessary rotations of the coordinate system make the times of order $O(n^3)$. The complex method could only solve Problem 3.9 for $n = 3$; upwards of $n = 10$ it no longer converged.

The last problem, Problem 3.10, which also has inequality constraints, turned out to be extremely difficult for all the search methods in the test. The main problem is one of scaling. Convergence in the neighborhood of the minimum can be achieved if, and practically only if, the step lengths in the coordinate directions are individually adjustable. They have to differ from each other by several powers of 10. For $n = 30$, no strategy managed to solve the problem within the maximum allowed computation time. The complex method sometimes failed to end the search within this time for $n = 10$. The intermediate results achieved after 8 hours are presented in Appendix A, Section A.3. All of the evolution strategies do better than the methods of Rosenbrock and Box.

The result that the two membered evolution strategy came closer to the objective than the multimembered evolution without recombination was not completely unexpected, because considerably fewer generations than mutations can occur within the allowed time. What is more surprising is that the $(10\,,100)$ strategy with recombination does almost as well as the two membered version. Here once again, the degree of freedom gained by the possibilities of recombination shows itself to advantage. The variances of the mutation step lengths do adjust themselves individually quite differently according to the situation and thus permit much faster convergence than with equal variances for all variables. The other evolution strategies only come as close as they do to the solution because the variances reach their relative lower bounds at different times, whereby differences in their sizes are introduced. This scaling process is, however, very much slower than the continuous process of adaptation brought about by the recombination mechanism.

6.4 Core storage required

Up to now, only the time has been considered as a measure of the computational cost. There is, however, another important characteristic that affects the applicability of optimization strategies, namely the core storage required. (Today nobody would use this term "core" here, but at the time these tests were performed, it was so called.) All indirect methods of quadratic optimization, which solve the linear equations for the extremal, require storage of order $O(n^2)$ for the matrix of coefficients. The same holds for quasi-Newton methods, except that here the significant rôle is played by the approximation to the inverse Hessian matrices. Most strategies that perform line searches in other than coordinate directions also require $O(n^2)$ words for the storage of n vectors, each with n coefficients. An exception to this rule is the conjugate gradient method of Fletcher and Reeves, which at each stage only needs to retain the latest generated direction vector for the subsequent iteration. Of the direct search methods included in the tests, the coordinate methods, the method of Hooke and Jeeves, and the evolution strategies work with only $O(n)$ words of core storage. How important the formal storage requirement of an optimization method can be is shown by the maximum number of variables for the tested strategies in Table 6.2. The limiting values range from 75 to 4,000 under the given conditions. There exist, of course, tricks such as segmentation for enabling larger programs to be run on smaller machines; the cost of the strategy should then take into account, however, the extra cost in preparation time for an optimization. (Here again, modern virtual storage techniques and the relative cheapness of memory chips make the considerations above look rather old-fashioned.)

In the following Table 6.11, all the strategies compared are listed again, together with the order of magnitude of their required computation time as obtained from the first set of tests (columns 1 and 2). The third column shows how the computation time would vary if each function call performed $O(n^2)$ rather than $O(n)$ operations, as would occur for the worst case of a general quadratic objective function. The fourth column gives the storage requirement, again only as an order of magnitude, and the fifth displays the product of the time and storage requirements from the two previous columns. Judging by the computation computation time alone, the variable metric strategy seems the best suited for true quadratic problems. In the least favorable case, however, it is more expensive than an indirect method and only faster in special cases. Problems having a very simple structure (e.g., Problem 1.1) can be solved just as well by direct search methods; the time they take is at worst only a constant factor more than that of a second order method.

If the total cost is measured by the product of time and storage requirements, all those strategies that store a two dimensional array of data, show up badly at least for problems with many variables. Since the coordinate methods have shown unreliable convergence, the method of Hooke and Jeeves and the evolution strategies remain as the least costly optimization methods. Their cost does not exceed that of indirect methods. The product of time and storage is not such a bad measure of the total cost; in many computing centers jobs have been, in fact, charged with the product of storage requested in K words and the time in seconds of occupation of the central processing unit (K-core-sec).

A comparison of the two membered and multimembered evolution strategies seems clearly to favor the simpler method. This is not surprising as several individuals in the multimembered procedure have to find their way towards the optimum. In nature, this process runs in parallel. Already in the early 1970s, first efforts towards constructing multi-processor computers were undertaken (see Barnes et al., 1968; Miranker, 1971).

Table 6.11: The dependence of the total costs of the search methods on the number of variables (n)

Strategy	Computation time for Problem 1.1	Computation time for Problem 1.2	Computation time for gen. quadr. probl.	Core storage	K-core-sec
FIBO,GOLD,LAGR	n^2	$n^{3\dagger}$	n^4	n	n^5
HOJE	$> n^2$	n^3	n^4	n	n^5
DSCG	n^2	n^4	n^4	n^2	n^6
DSCP	n^2	n^3	n^4	n^2	n^6
POWE	n^2	$n^{3\dagger}$	n^4	n^2	n^6
DFPS	n^2	$n^{2.5}$	$n^{3.5}$	n^2	$n^{5.5}$
SIMP	$> n^3$	n^5	n^5	n^2	n^7
ROSE	n^3	n^4	n^4	n^2	n^6
COMP	$> n^3$	$n^{5\dagger}$	n^5	n^2	n^7
EVOL,GRUP,REKO	n^2	n^3	n^4	n	n^5

On such a parallel computer, supposing it had 100 sub-units, one could simultaneously perform all the mutations and objective function evaluations of one generation in the $(10\,,100)$ evolution strategy. The time required for the optimization would be about two orders of magnitude less than it is with a serially operating machine. In Figures 6.14 and 6.15 the dotted lines show the results that would be obtained by the $(10\,,100)$ strategy without recombination in the hypothetical case of parallel operation. No other methods can make use of parallel operations to such an extent. On SIMD (single instructions, multiple data) architectures, the possible speedup is sharply limited by the percentages of a program's scalar and vector operations. Using array arithmetic for all matrix and vector operations, the execution time of a program may be accelerated at most by a factor of five, given that these operations would serially take 80% of the computation time. On MIMD (multiple instructions, multiple data) machines, the speedup is limited by the number of processing units a program can make use of and by the amount of communication needed between the processors and the data store(s). Most classical optimization algorithms cannot economically employ large MIMD computers–even the less, the more sophisticated the procedures are. Multimembered evolution strategies, however, are easily scalable to any number of processors and communication links between them. For a taxonomy of parallel versions of evolution strategies, see Hoffmeister and Schwefel (1990).

†Not sure to converge