Large-scale nonlinear constrained optimization: a current survey

A. R. Conn, N. I. M. Gould and Ph. L. Toint

January 16, 2018

Abstract

Much progress has been made in constrained nonlinear optimization in the past ten years, but most large-scale problems still represent a considerable obstacle.

In this survey paper we will attempt to give an overview of the current approaches, including interior and exterior methods and algorithms based upon trust regions and line searches. In addition, the importance of software, numerical linear algebra and testing will be addressed. We will try to explain why the difficulties arise, how attempts are being made to overcome them and some of the problems that still remain.

Although there will be some emphasis on the LANCELOT and CUTE projects, the intention is to give a broad picture of the state-of-the-art.

1 Introduction

We shall first state the most general form of the problem that we are addressing, namely

$$\underset{x\in\Re^n}{\text{minimize}} \quad f(x) \tag{1.1}$$

subject to the general (possibly nonlinear) inequality constraints

$$c_j(x) \leq 0, \quad 1 \leq j \leq l, \tag{1.2}$$

to the (possibly nonlinear) equality constraints

$$c_j(x) = 0, \quad l+1 \leq j \leq m, \tag{1.3}$$

and the simple bounds

$$l_i \leq x_i \leq u_i, \quad 1 \leq i \leq n. \tag{1.4}$$

Here, f and the c_j are all assumed to be twice-continuously differentiable and any of the bounds in (1.4) may be infinite.

We only expect to obtain local minimizers. This presents no problems in convex programming, where all local minima are indeed global (for example, in linear programming), but even for small, general nonlinear programming problems it is usually extremely difficult to verify globality. For large problems, with current techniques it is practically impossible. Fortunately, in many situations, an algorithm that determines local optima suffices.

Our primary interest here is in problems that involve a large number of variables and/or constraints. Consequently, it seems worthwhile to elaborate as to what we mean by large. Firstly, this notion is clearly *computer dependent*. Secondly, the notion of size is *problem dependent*. A highly nonlinear problem in one hundred variables could be considered large, whereas in linear programming it is possible to solve problems in five million variables. Similarly, it also depends upon the *structure of the problem*. Many large-scale nonlinear problems arise from the modeling of very complicated systems that may be subdivided into loosely connected subsystems. This structure may often be reflected in the mathematical formulation of the problem and exploiting it is often crucial if one wants to obtain an answer efficiently. The complexity of the structure is often a key factor in assessing the size of a problem. In addition, the notion of a large problem depends upon the *frequency* with which one expects to solve a particular instance or closely related problem. When one anticipates solving the same class of problems many times, one can afford to expend a significant amount of energy analyzing and exploiting the underlying structure.

Thus, although it is not possible to state categorically that a problem in, say, seven hundred variables is large, suffice it to say that, today, a problem in fifty variables is small and a generally nonlinear problem in five thousand variables and one thousand nonlinear constraints is large.

At the risk of stating the obvious, the world is not linear and *accurate* modeling of physical and scientific phenomena often leads to large-scale nonlinear optimization. In our opinion, the frequent use of linear models is not an indication that nonlinear problems do not abound. Rather, it is a statement of the desire to use an algorithm (the simplex method) that is readily understood and is well-known to be suitable for large problems. We would like to convince you that you should consider solving nonlinear programs when they are more appropriate. It should be emphasized that solutions to large nonlinear problems on moderate workstations in a reasonable amount of time are currently quite possible. Furthermore, in practice one is often only seeking improvement rather than assured optimality (another reason why local solutions may suffice). This fact makes problems that at first sight seem impossible (for example, control problems that one wishes to solve in something like real-time), tractable.

In the past twenty years rather sophisticated and reliable techniques for small-scale problems have been developed (see Chapters 1 and 3 of Nemhauser *et al.*, 1989, and the chapters of Bartholomew-Biggs and Fletcher in this volume). However, efficient algorithms for small-scale problems do not necessarily translate into efficient algorithms for large-scale problems (see, for example, Bartholomew-Biggs and Hernandez, 1994). Thus, it is not adequate to take existing optimization software for small problems and apply it to large ones, hoping that the increased capacity in computing will take care of the growth in problem size. By contrast, we could expect that an efficient method for large-scale problems be at least moderately efficient for small-scale problems.

Notwithstanding, it is essential to know and understand the small-scale background.

Without a doubt, the availability of powerful workstations and supercomputers (both parallel and sequential) has encouraged research in algorithms for large-scale problems, but the main reason we can solve very large problems is because we can *exploit structure*. Moreover, the state-of-the-art of large-scale nonlinear programming has progressed so much in the past decade that it is reasonable to ask the question 'Is it worthwhile to design algorithms that are unsuitable for large-scale problems?' At present, the answer is likely to be in the affirmative, for example, for problems where the cost of function evaluations is very high or for problems with extremely nonlinear behaviour and/or difficult scaling.

Furthermore, some of the more mundane tasks, such as the input of problems, are important and non-trivial issues. The evaluation of results is even more important and difficult. The scope of some of the problems tackled by LANCELOT and included in CUTE (see below for more details on these two packages) contains a large number of nonlinear optimization problems of various sizes and difficulty, representing both 'academic' and 'real world' applications. Both constrained and unconstrained examples are included. The problems we have solved to date using LANCELOT range from problems with 20,000 variables and 10,000 nonlinear constraints to small problems with less than 10 variables and constraints. It is worth mentioning that some of the most difficult problems are small (for example, LANCELOT has been unable to solve a problem with 149 variables, a quadratic objective function and 100 nonlinear constraints). It is also worth stating that although LANCELOT was designed with large-scale problems in mind, it is very suitable for solving small-scale problems.

Of course, there are many details that can contribute to the difficulty of a problem. Unfortunately, none of us are good at handling them all. Scaling is a well-known difficulty for which one has methods to try, but it is clear that we would like to be able to do much better. There are approaches that are usually effective in handling indefiniteness, but here again one feels that these are far from ideal. Both primal and dual degeneracy are often perceptible as difficulties. It is not always clear as to how they can best be tackled.

There is a very real difficulty associated with the fact that many practitioners prefer good solutions to bad models rather than less good solutions to more accurate (and thus better and probably more complicated) models. Indeed a related problem that has frequently been an unwelcome accompaniment to nonlinear optimization is that the user of the software needs to be relatively sophisticated.

Finally, there are all the problems related to solving systems of symmetric linear equations, since this is, in many ways, the kernel computation in nonlinear optimization.

2 Basic Background

Although much of the fundamental background is covered in this volume by the contributions of Bartholomew-Biggs (1994) and Fletcher (1994), there are some very basic comments that relate to large-scale optimization that we would like to mention here.

Firstly, the most basic approach to unconstrained optimization is undoubtedly steepest descent. ¿From the point of view of storage, this is a splendid method for large-scale optimization. However, it is intolerably slow since its convergence rate is linear with a rate constant that may be uncomfortably close to one. The other extreme is a safeguarded Newton's method, which has a second-order convergence rate. But in this case, the standard implementation requires too much storage ($O(n^2)$) and too much work per iteration ($O(n^3)$ flops). In fact, much of what we need to concern ourselves with is how to do as little as possible initially (steepest-descent-like) and enough eventually to guarantee an acceptable convergence rate (Newton-like). In effect, this is the standard problem of global versus asymptotic behavior, since the weak behavior of steepest descent is enough to guarantee global convergence (convergence to a stationary point from any starting point).

One effective technique for large structured problems (mentioned in Section 4 of Fletcher, 1994) is intelligent finite differencing (originally due to Curtis *et al.*, 1974). However, the standard steepest descent/Newton's method compromise is quasi-Newton methods. Once again details are given in Fletcher (1994), but essentially the idea is to use low rank updates to an initial approximation to the Hessian matrix (usually a (scaled) identity matrix). These methods possess a sufficiently fast (superlinear) convergence rate. The updates can be posed as minimization problems. For example, PSB (see Powell, 1970) may be determined from

$$\underset{U \in \Re^{n^2}}{\text{minimize}} \quad \|U\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} U_{ij}^2 \tag{2.1}$$

subject to the quasi-Newton equations

$$\begin{array}{rcl} U\delta & = & \gamma - B\delta \\ U & = & U^T, \end{array} \tag{2.2}$$

where

$$\begin{array}{rcl} B^+ & = & B + U \\ \delta & = & x^+ - x \\ \gamma & = & g^+ - g. \end{array} \tag{2.3}$$

Here the superscript + indicates an update, B is the Hessian approximation and g is the gradient of f.

A natural extension to structured problems is to impose sparsity by considering (see Toint, 1981a)

$$\underset{U \in \Re^{n^2}}{\text{minimize}} \quad \|U\|_F^2 \tag{2.4}$$

subject to the constraints

$$\begin{array}{rcl} U\delta & = & \gamma - B\delta \\ U & = & U^T \\ \text{and } U_{ij} & = & 0, ij \in S, \end{array} \tag{2.5}$$

where S specifies the sparsity pattern.

Unfortunately this approach has not turned out to be very successful in practise (see Sorensen, 1981). On the other hand, the quasi-Newton approach can be successfully applied to large-scale problems if the partially separable structure of the problem (see below) is taken into account. If quasi-Newton methods are preferred to exact second derivatives[1], it is thus possible to approximate the Hessian of each element function f_i individually, using a secant equation of the type (2.3) for each one of them. This technique is called 'partitioned updating' and was introduced by Griewank and Toint (1982b). This technique is substantially more successful than the sparse updating method just described and is provided as an option within the LANCELOT package.

Another compromise between steepest descent and Newton's method is the method of conjugate directions. In the large-scale case we tend to think of it as closer to steepest descent, but in some contexts (good preconditioners, for example) it may be closer to Newton's method. Steepest descent with the inverse of a positive definite Hessian as preconditioner is indeed Newton's method.

Conjugate direction methods maintain finite Q-convergence (that is, converge for a positive definite quadratic problem in a finite number of iterations; no more than n, the dimension of the space). This is not really very relevant for large-scale problems, where n is large. One can think of conjugacy as a generalization of orthogonality[2]. Thus it is not that surprising that these directions can be derived via Gram-Schmidt orthogonalization, either as three-term recurrences or using Lanczos orthogonalization, although some care has to be taken to make the process numerically stable (see, for example, Golub and Loan, 1989, Chapter 9). As a consequence of these recurrences, conjugate direction algorithms can be implemented storing only a few vectors (three to five, depending on the precise method used). With exact line searches and exact arithmetic, the method is n-step superlinearly convergent, in general. The proof depends critically upon restarts — otherwise convergence is linear. In practise they nearly always converge linearly, but for large n, n-step superlinear is not much better. Of course, what one wants is a fast linear rate, which preconditioning can achieve.

If we think of Lanczos as

$$Q^T A Q = T, \tag{2.6}$$

where T is tridiagonal and Q is the matrix whose columns are the Lanczos vectors, then the process can also be carried out in block form with T block

[1] In our experience, this is very seldom necessary.

[2] A set of directions $\{d_i\}_1^k$ are G-conjugate for positive definite G if and only if $d_i^T G d_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta.

triangular. One can then work with the blocks separately and exploit a parallel environment (Nash and Sofer, 1991).

Another way to reduce storage is to use limited memory methods. For example Liu and Nocedal (1989) (see also Liu and Nocedal, 1988), use an inverse BFGS update in the form

$$B^+ = V^T B V + \rho \delta \delta^T, \tag{2.7}$$

where $\rho = 1/\gamma^T \delta$ and $V = I - \rho \gamma \delta^T$. The basic idea is to start with a B that can be stored efficiently, for example a scaled version of the identity matrix. One then updates m times, however without storing the updated matrices *explicitly* but instead storing the m pairs γ and δ. Most importantly m is typically very small, say five. The scaling of the initial matrix is also important. Other recent references include Byrd *et al.* (1993) and Zou *et al.* (1993).

However, it is unclear as to whether the relative success of naive preconditioners, limited memory with small m and naive scaling of the identity matrix are mostly a consequence of the not very extensive testing that has been carried out to date. In particular, most problems tested seem to be rather well scaled.

2.1 Solving the Linear System

Typically, the major computational task in optimization is to solve a system of linear equations that arises from the fact that one uses quadratic models and stationary points are characterized by gradients being zero. In addition, optimality conditions and/or reduced methods for constrained problems give rise to (generalized) least squares problems and linear systems involving the Karush-Kuhn-Tucker matrix. Thus progress in solving large linear systems has implications for large-scale optimization (see, for example, the contribution of Björck, 1994, in this volume). If the system is written

$$Bd = -g \tag{2.8}$$

then ideally, we would like to combine the determination of B with the solution of (2.8). If possible we would choose the matrix of exact second derivatives (maybe in a reduced space) for B. As we will see later, this often can be done if structure is suitably exploited. The linear system can then be solved using direct or iterative techniques.

We first consider direct methods. There are two main approaches, namely multifrontal techniques and sparse Cholesky factorizations. Very briefly, the former approach tries to assemble the required entries in a piecemeal manner. Once a complete column and row are assembled one can do the corresponding elimination, thus building up the corresponding elements of L and U. For details see Conn *et al.* (1993a), Duff *et al.* (1986, Chapter 10), Duff *et al.* (1988), Duff and Reid (1982), Duff and Reid (1983) and Duff and Reid (1993). By contrast, the sparse Cholesky factorization primarily tries to order the rows and columns of B whilst maintaining reasonable stability by including the possibility of adding appropriate quantities to the diagonals of B, if necessary, (Chapter 3 of Conn *et al.*, 1992b, Gill and Murray, 1974, Gill *et al.*, 1992, Schlick, 1993 and Schnabel and Eskow, 1991). For example, Schnabel and Eskow use Gerschgorin bounds to determine the amount to add to the diagonal. They choose diagonal pivots and change the diagonal as little as is reasonable in order to maintain sufficient positive definiteness. All the proposed methods use about $O(n^2)$ additional work as compared with standard Cholesky. It is interesting to remark that these methods are related to l_2 trust region/Levenberg Marquardt algorithms, although the latter are using a rank n update rather than the normally considerably lower rank updates used above, (Hebden, 1973, Levenberg, 1944, Marquardt, 1963 and Moré, 1978).

The iterative method of choice is that of (preconditioned) conjugate gradients. Thus we need to solve (2.8), where B is a (possibly perturbed) approximation to the Hessian matrix $\nabla_{xx} f$. The perturbation may be obtained as the conjugate gradient algorithm proceeds in what we think is an elegant way that preserves conjugacy, see Arioli *et al.* (1993).

3 Some Existing Methods

Let us first consider the most venerable and best known nonlinear optimization algorithm that was designed with large-scale problems in mind. The origins of MINOS (Murtagh and Saunders, 1987) come from Robinson (1972) and Rosen and Kreuser (1972). The method can be considered to be an extension of the simplex method, since both are a reduced gradient technique. Thus the structure exploited is *sparsity* and the essential technology used is closely related to the linear programming technology of the simplex method. In particular, MINOS replaces

$$\begin{array}{ll} \underset{x\in\mathbf{R}^n, y\in\mathbf{R}^m}{\text{minimize}} & F(x) + c^T x + d^T y \\ \text{subject to} & f(x) + A_1 y = b_1 \\ & A_2 x \ + A_3 y = b_2 \\ \text{and} & l_x \le x \le u_x \\ & l_y \le y \le u_y \end{array} \tag{3.1}$$

with

$$\underset{x\in\mathbf{R}^n, y\in\mathbf{R}^m}{\text{minimize}} \quad F(x) + c^T x + d^T y + \lambda_k^T (f(x) - \tilde{f}(x)) + \tfrac{1}{2}\rho (f(x) - \tilde{f}(x))^T (f(x) - \tilde{f}(x))$$

$$\begin{array}{ll} \text{subject to} & \tilde{f}(x) + A_1 y = b_1 \\ & A_2 x + A_3 y = b_2 \\ \text{and} & l_x \le x \le u_x \\ & l_y \le y \le u_y, \\ \text{where} & \tilde{f}(x) = f(x_k) + J_k (x - x_k), \end{array} \tag{3.2}$$

and J_k denotes the Jacobian of f evaluated at x_k. We note that the nonlinear contributions to the constraints are linearized. One then formulates a quadratic model for the corresponding augmented Lagrangian objective function (see Fletcher, 1987, Section 12.2). Writing the activities that are determined by the general linear constraints as

$$\hat{A}x = \begin{pmatrix} B & S & N \end{pmatrix} x = b, \tag{3.3}$$

a basis for the null space is given by the columns of the matrix Z, where

$$Z^T = \begin{pmatrix} -[B^{-1}S]^T & I & 0 \end{pmatrix}. \tag{3.4}$$

This follows directly from the fact that

$$\hat{A}Z = 0 \text{ and } (0 \quad 0 \quad I)Z = 0. \tag{3.5}$$

Since most of the computation in the outline above involves the inverse of the basis matrix, B^{-1}, it is hardly surprising that exploitation of structure in this algorithm mimics exploitation of the same structure in the simplex method.

More recent methods that are closely related to sequential quadratic programming (see Bartholomew-Biggs, 1994, Section 5, for a general description) are what Gill *et al.* (1993b) call transformed Hessian methods (see also Eldersveld, 1992). Thus consider the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{3.6}$$

subject to

$$c_j(x) \geq 0, \quad 1 \leq j \leq l, \tag{3.7}$$

and the positivity constraints

$$x_i \geq 0 \quad 1 \leq i \leq n. \tag{3.8}$$

They then try to find $(\delta_x, \delta_\lambda)$ by minimizing a *quadratic* model of the Lagrangian subject to a *linear* model of the constraints (3.7). For large problems the efficiency of the linear algebra required to solve the created quadratic program is crucial. One has to repeatedly solve a linear system with the Karush-Kuhn-Tucker matrix

$$\begin{pmatrix} B^{(k)} & A_\omega^T \\ A_\omega & 0 \end{pmatrix}, \quad \text{where } A_\omega = \begin{pmatrix} A \\ I \end{pmatrix}. \tag{3.9}$$

It is worth remarking that solving such systems has general applicability to problems with linear constraints (see, for example, Arioli *et al.*, 1993 and Forsgren and Murray, 1993). Gill, Murray and Saunders use generalized TQ factorizations with

$$A_\omega Q = (0 \; T) \text{ and } Q^T H Q = R^T R. \tag{3.10}$$

Now, the Hessian H required for the gradient of the quadratic program's objective function can be determined from

$$H = Q^{-T} R^T R Q^{-1}. \tag{3.11}$$

The solution to the quadratic program is completely determined by the upper triangular matrix T, the matrix Q and the first $n - t$ rows of the upper trapezoidal matrix R. If we let Z denote the first $n - t$ columns of Q and call the remaining columns of Q, Y, then $Z^T HZ$ is the usual *reduced Hessian* and $Q^T HQ$ is the *transformed Hessian.* Furthermore, Y spans the range space associated with A_ω.

In order to avoid changing A_ω, one adds slacks explicitly and the trick is to choose Q's that are relatively easily invertible, because of the need for (3.11). Moreover, only a part of R need be stored and one can arrange not to lose the structure in H that results from the additional slacks by permuting A_ω appropriately. One can think of this as being a non-orthogonal (and thus appropriate for large-scale) version of NPSOL (Gill *et al.*, 1986).

The above approaches are line-search based. There are also excellent algorithms that are trust-region based. Once again these are mentioned in Fletcher (1994, Section 1) and further details and references are given in Moré (1983). Consider first the unconstrained problem.

The salient features we wish to recall here is that one uses a suitable *model* for the objective[3] that one *trusts* over a suitable region[4]. One then compares the actual reduction with the predicted reduction. If the comparison is sufficiently favorable, the trust region is expanded and the current point is updated. If it is sufficiently unfavorable, the trust region is reduced and the current point is unchanged. Otherwise, only the current point is updated. Continuity guarantees that eventually reduction of the trust region must ensure that the predicted reduction is close enough to the actual reduction, which in turn guarantees that the trust region is bounded away from zero. Global convergence is assured as long as we do as well as minimizing the model, within the trust region, along the steepest descent direction (which defines the Cauchy point). Eventually the trust region is irrelevant, which guarantees a fast asymptotic convergence rate as long as the underlying model optimization is suitably chosen (for example, a safe-guarded Newton-like method).

The generalization to simple bounds is straightforward. For example, if one uses the l_∞ norm, then the trust region is a box. The feasible region corresponding to simple bounds is also a box. The intersection of two boxes is a box. One now defines a generalized Cauchy point as the minimum along the *projected* gradient path within the trust region, where the projection is with respect to the simple bounds. Since we are dealing with boxes the projection is trivial. Such a projected gradient approach was proposed by McCormick (1969), and independently in Bertsekas (1982) and Levitin and Polyak (1966). More recently it has been exploited extensively in the context of large-scale optimization by many authors, see for example Conn *et al.* (1988b), Dembo and Tulowitski (1983), Moré and Toraldo (1989), and Moré and Toraldo (1991). As in the unconstrained case, global convergence can be guaranteed, provided one does at least as well as the generalized Cauchy point. One obtains better

[3] e.g., a quadratic model given by the second-order Taylor's expansion about the current point

[4] e.g., a sphere or box

convergence, and ultimately a satisfactory asymptotic convergence rate, by further reducing the model function. This is the trust region basis for the kernel algorithm SBMIN (Conn *et al.*, 1988a) of LANCELOT (Conn *et al.*, 1992b). It can be summarized as follows:

- Find the generalized Cauchy point based upon a local (quadratic) model.
- Fix activities to those at the generalized Cauchy point.
- (Approximately) solve the resulting reduced problem whilst maintaining account of the trust region and bounds.
- Determine whether the current point is acceptable and update the trust region radius accordingly.

The supporting theory in Conn *et al.* (1988a) verifies that the algorithm converges to a first-order stationary point, provided the quadratic model is reasonable. Moreover, the correct activities are identified after a finite number of iterations if strict complementarity[5] is satisfied and the activities determined by the generalized Cauchy point are kept active when the model is further reduced in the inner iteration.

What makes this approach particularly attractive for large-scale problems is that the determination of the generalized Cauchy point is easy (and need not be exact) and one can use suitable *unconstrained* large-scale techniques. An example would be truncated, preconditioned conjugate gradients (see, for example, Steihaug, 1983a, Steihaug, 1983b and Toint, 1981b). Furthermore, often one is able to exploit the structure in order to use exact second derivatives (see below). Usually one never needs the Hessian matrix (or its approximation) but rather the corresponding matrix-vector products. Here again it is possible to exploit structure. The standard structure to exploit is sparsity and this is basic to large-scale numerical linear algebra, see for example Duff *et al.* (1986) and George and Liu (1981). In addition, most of the improvements in the simplex method have depended upon such exploitation. LANCELOT exploits a more general form of structure. The basic idea was first introduced in Griewank and Toint (1982a). We introduced a slight generalization, exploiting this very pervasive type of structure, which we call *group partial separability.* Consider two different functions, $f_1(x) = x_{50}^4$ and $f_2(x) = \left[\sum_{i=1}^{5,000,000} x_i\right]^4$, where $x \in \Re^{5,000,000}$. We first note that $\nabla_{xx} f_1$ is very sparse[6] and $\nabla_{xx} f_2$ is completely dense. However, the important structure to note is that both functions have an invariant subspace of dimension $4,999,999$. If we use the linear transformation $w = e^T x$, where e is the vector of ones, then $f_2(x)$ is transformed to w^4. Imagine having sums of such functions, not necessarily independent. Then you have the fundamental idea. Moreover, it is not unusual to have many similar f_i's with just different labellings. In fact the economies of storage are such that often one is able to solve quite large problems on small machines.

A function $f(x)$ is said to be *group partially separable* if:

[5] The case where strict complementarity fails to hold is considered by Lescrenier (1991).

[6] It has only one non-zero entry.

1. the function can be expressed in the form

$$f(x) = \sum_{i=1}^{n_g} g_i(\alpha_i(x)); \tag{3.12}$$

2. each of the *group functions* $g_i(\alpha)$ is a twice continuously differentiable function of the single variable α;

3. the function

$$\alpha_i(x) = \sum_{j \in \mathcal{J}_i} w_{i,j} f_j(x^{[j]}) + a_i^T x - b_i \tag{3.13}$$

   is known as the i-th *group*;

4. each of the index sets $\mathcal{J}_i$ is a subset of $\{1, \ldots, n_e\}$;

5. each of the *nonlinear element functions* f_j is a twice continuously differentiable function of a subset $x^{[j]}$ of the variables x. Each function is assumed to have a large invariant subspace. Usually, this is manifested by $x^{[j]}$ comprising a small fraction of the variables x;

6. the gradient a_i of each of the *linear element functions* $a_i^T x - b_i$ is, in general, sparse; and

7. the $w_{i,j}$ are known as element *weights*.

An additional degree of freedom may be present in a partially separable structure. Often a distinction can be made between the *elemental* variables (the problem's variables that effectively occur in the expression of the considered element) and *internal* variables associated with a given element[7]. A more thorough introduction to group partial separability is given by Conn *et al.* (1990a). SBMIN assumes that the objective function $f(x)$ is of this form.

To summarize, we now know that LANCELOT is trust-region based, uses SBMIN as its kernel algorithm and exploits structure via group partial separability. We now explain how it is extended to handle general equality constraints. Inequalities are changed to equalities by the addition of slacks. Like MINOS it uses the augmented Lagrangian, which we can think of as a Lagrangian with additional quadratic (exterior) penalty terms.

The objective function and general constraints are combined into a composite function, the *augmented Lagrangian function*,

$$\Phi(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i=1}^{m} c_i(x)^2, \tag{3.14}$$

where the components λ_i of the vector λ are known as *Lagrange multiplier estimates*, and μ is known as the *penalty parameter*.

The constrained minimization problem (1.1), (1.3) and (1.4) is now solved by finding approximate minimizers of Φ for a carefully constructed sequence of

[7] For example, in f_2 above, one could consider w to be an internal variable and the x's to be the elemental variables.

Lagrange multiplier estimates, constraint scaling factors and penalty parameters.

The approach can be summarized as

- Test for convergence.
  Convergence occurs when the iterate is sufficiently stationary (i.e., the projected gradient of the augmented Lagrangian with respect to the simple bounds is small enough) and the current approximate minimizer of Φ is sufficiently feasible.

- Major iteration.
  Use the simple bounds algorithm SBMIN to find a sufficiently stationary approximate minimizer of Φ considered as a function of x and constrained explicitly by the simple bounds.

- Choice of update.
  If sufficiently feasible, *update multipliers* and *decrease* the tolerances for feasibility and stationarity.
  Otherwise, *decrease the penalty parameter* and *reset* the tolerances for feasibility and stationarity.

We use first-order updates for the multipliers, namely

$$\lambda_i^+ = \lambda_i + c_i(x^k)/\mu. \tag{3.15}$$

Reset and update rules for the multipliers, stationarity, feasibility and the penalty parameter are all analyzed in the theory of Conn *et al.* (1991) and Conn *et al.* (1992d). There we are able to show that under suitable conditions we converge to a first-order stationary point for the nonlinear programming problem. Furthermore, if we have a single limit point, we eventually stop reducing the penalty parameter, μ. Finally, under somewhat stronger conditions, one ultimately requires only a single iteration of the simple bounds algorithm to satisfy stationarity for the outer iteration. This, plus many options, is the state-of-the-art of LANCELOT A.

4 A Testing Environment

It is not that astonishing that during our research we were soon led to the frustrating question of testing and evaluating algorithms for large-scale nonlinear optimization. Moreover, there is a rapid appreciation of how difficult this task is — hence the dearth of published nonlinear results obtained with MINOS, even though it has been available for over fifteen years.

The origin of our so-called standard input format (SIF) in LANCELOT was that the setting up of test problems that accounted for the group partially separable structure was tremendously tiresome. Group partial separability simplifies the optimization but complicates the input. Conn *et al.* (1992b, Chapter 2) provide an introduction to the SIF, including the considerations given to its design. Chapter 7 of the same reference serves as a detailed manual on the format.

Additional requirements in a suitable testing environment include

- a large database of test problems and a means of managing it,
- the ability to compare results with the best of the existing optimization packages,
- facilities to test algorithmic ideas on the collection of problems, and finally
- making this all freely available to the community.

Hence the **C**onstrained and **U**nconstrained **T**esting **E**nvironment of CUTE (Bongartz *et al.*, 1993). This offers a large growing database of test problems written in SIF. The test set covers, amongst others,

- the 'Argonne test set' (Moré *et al.*, 1981), the Testpack report (Buckley, 1989), the Hock and Schittkowski collection (Hock and Schittkowski, 1981), the Dembo network problems (Dembo, 1984), the Moré-Toraldo quadratic problems (Moré and Toraldo, 1991), the Boggs-Tolle problems (Boggs and Tolle, 1989), the Toint-Tuyttens network model problems (Toint and Tuyttens, 1990), and Gould's quadratic programming problems (Gould, 1991),
- most problems from the PSPMIN collection (Toint, 1983),
- problems inspired by the orthogonal regression report by Gulliksson (Gulliksson, 1990),
- some problems from the Minpack-2 test problem collection (Averick *et al.*, 1991, Averick and Moré, 1991) and from the second Schittkowski collection (Schittkowski, 1987) and
- a large number of original problems from a variety of application areas.

Each problem comes with a classification listing the type of problem, degree of available derivatives, origin and size. There are tools provided to create, maintain and update the classification database and also to select problem SIF files on the basis of the classifications. Furthermore, we realize that not everyone, especially non-users of LANCELOT, is equally enthusiastic about using partial separability and the SIF. However, the database of test problems provided by CUTE is clearly very useful. Thus CUTE provides tools to allow an interface between problems, specified using the SIF, and other existing nonlinear programming packages, in addition to providing a relatively easy means of building interfaces with new algorithms. When applicable these tools are provided in sparse and dense formats.

At the present time, interfaces are available for the following:

- **MINOS (see above)**
  We currently have interfaces for MINOS 5.3 and MINOS 5.4.
- **NPSOL of Gill *et al.* (1986)**
  This package is designed to minimize smooth functions subject to constraints, which may include simple bounds, linear constraints, and smooth

nonlinear constraints. The software uses a sequential quadratic programming algorithm, where bounds, linear constraints and nonlinear constraints are treated separately. Unlike MINOS, NPSOL stores all matrices in dense format, and is therefore not intended for large sparse problems.

- **OSL of International Business Machines Corporation (1990)**
This package obtains solutions to quadratic programming problems where the Hessian matrix is assumed positive semidefinite. It is intended to be suitable for large-scale problems.

- **TENMIN of Schnabel and Chow (1991)**
This package is intended for problems where the cost of storing one n by n matrix (where n is the number of variables), and factoring it at each iteration, is acceptable. The software allows the user to choose between a tensor method for unconstrained optimization, and an analogous standard method based upon a quadratic model. The tensor method bases each iteration upon a specially constructed fourth-order model of the objective function that is not significantly more expensive to form, store, or solve than the standard quadratic model.

- **UNCMIN of Koontz *et al.* (1985) that corresponds closely to the pseudocode in Dennis and Schnabel (1983)**
This package is designed for unconstrained minimization and has options that include both line search and trust region approaches. The provided options include analytic gradients or difference approximations with analytic Hessians or finite difference Hessians (from analytic or finite difference gradients) or secant methods (BFGS).

- **VA15 of Liu and Nocedal (1989)**
This package solves general nonlinear unconstrained problems using a limited memory BFGS method. It is intended for large-scale problems.

- **VE09 of Gould (1991)**
This package obtains local solutions to general, non-convex quadratic programming problems, using an active set method, and is intended to be suitable for large-scale problems.

- **VE14 of Conn *et al.* (1993g)**
This package solves bound-constrained quadratic programming problems using a barrier function method and is again intended to be suitable for large-scale problems.

- **VF13 of Powell (1982)**
This package solves general nonlinearly constrained problems using a sequential quadratic programming technique.

VA15, VE09, VE14 and VF13 are part of the Harwell Subroutine Library (1993).

5 Further Developments

Having described LANCELOT A, we now consider future developments. Firstly it is obvious that we would like to learn from our experiences with LANCELOT A, but this is not necessarily easy. Unfortunately one soon discovers that one should do a great deal of testing, including experience with the best competitive algorithms on the same non-trivial problems. However one also discovers that (fortunately, occasionally) relatively innocuous seeming changes, like changing the initial trust region size from one to two, may change the solution time by several orders of magnitude. A more detailed example of the difficulties of definitive testing is illustrated by the following tale. Amongst our many applications we have some in structural optimization that give rise to minimax problems which, when posed as nonlinear programming problems, contain very many more inequality constraints than variables (see, for example, Achtziger *et al.*, 1992). Consequently if they are solved via LANCELOT A it is necessary to add very many slack variables. In fact the particular incidence we have in mind involved a discrete plate problem[8] with 343 variables and 8,958 inequality constraints. Thus, with the addition of slacks, one has a problem in 9,301 variables and 8,958 equality constraints. The run we made with the LANCELOT default parameters took 117 hours on an IBM RISC/6000 320 — not particularly encouraging! This provided one motivating factor for us to consider handling inequalities directly via barrier functions.

We now consider barrier functions and their extension in more detail. As is discussed in Bartholomew-Biggs (1994, Section 4), historically a shift was introduced to the quadratic penalty function to avoid updating the penalty parameter more than a finite number of times, thus giving the multiplier methods/augmented Lagrangian functions already used above. It seems reasonable to consider doing the same for logarithmic barrier functions and indeed in recent years there has been a flurry of activity in this area (Breitfeld and Shanno, 1993a, Breitfeld and Shanno, 1993b, Conn *et al.*, 1992a, Freund, 1991, Gill *et al.*, 1988, Jittorntrum and Osborne, 1980, Jensen and Polyak, 1993, Nash *et al.*, 1993, Polyak, 1992 and Powell, 1992).

To see the augmented Lagrangian as a shifted/modified quadratic penalty function we note that (3.14) is equivalent to $\hat{\Phi}(x, \lambda, \mu) - \frac{1}{2\mu}\sum_{i=1}^{m} s_i^2$, where

$$\hat{\Phi}(x, \lambda, \mu) = f(x) + \frac{1}{2\mu}\sum_{i=1}^{m} [c_i(x) + s_i]^2 , \tag{5.1}$$

and the *shifts* $s_i = \mu\lambda_i$. Note that if we assume that the λ_i are bounded, $\mu \to 0$ implies that $s_i \to 0$. But then we can say that the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{5.2}$$

subject to

$$c_i(x) + s_i = 0, \quad 1 \leq i \leq m, \tag{5.3}$$

[8] Known as HAIFAL.SIF in the CUTE distribution.

converges to

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{5.4}$$

subject to

$$c_i(x) = 0, \quad 1 \leq i \leq m, \tag{5.5}$$

as μ tends to zero. But (5.1) is the quadratic penalty function for (5.2) and (5.3), and the problem given by (5.2) and (5.3) is equivalent to

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{5.6}$$

subject to

$$\frac{1}{2s_i}[c_i(x) + s_i]^2 = 0, \quad 1 \leq i \leq m, \tag{5.7}$$

provided that $s_i \neq 0$. Now the classical Lagrangian for this latter formulation is

$$f(x) + \frac{1}{2\mu} \sum_{i=1}^{m} [c_i(x) + s_i]^2 = \hat{\Phi}(x, \lambda, \mu), \tag{5.8}$$

with $s_i = \mu\lambda_i$. Thus one can think of this as a Lagrangian quadratic penalty function.

Let us now consider a similar development for the logarithmic barrier function,

$$\Psi(x, \lambda, s) = f(x) - \mu \sum_{i=1}^{m} \log\left[c_i(x) + s_i\right], \tag{5.9}$$

corresponding to the problem

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{5.10}$$

subject to

$$c_i(x) \geq 0, \quad 1 \leq i \leq m. \tag{5.11}$$

Taking $\mu = \lambda_i s_i$ we rewrite this as

$$\Psi(x, \lambda, s) = f(x) - \sum_{i=1}^{m} \lambda_i s_i \log\left[c_i(x) + s_i\right], \tag{5.12}$$

with corresponding first-order update

$$\lambda_i^+ = \lambda_i s_i / \left[c_i + s_i\right]. \tag{5.13}$$

Analogously to the presentation above, Polyak points out that (5.11) is equivalent to

$$s_i \log\left[1 + c_i(x)/s_i\right] \geq 0, \quad 1 \leq i \leq m, \tag{5.14}$$

and the classical Lagrangian for the problem (5.10), subject to (5.14), is given by $\hat{\Psi}(x, \lambda, s)$, where

$$\hat{\Psi}(x, \lambda, s) = f(x) - \sum_{i=1}^{m} \lambda_i s_i \log\left[1 + c_i(x)/s_i\right]. \tag{5.15}$$

But then $\hat{\Psi} = \Psi - \mu \sum_{i=1}^{m} \log[s_i]$, and the last term is independent of x.

Gill *et al.* (1988) carried out their analysis for linear programs, chose $s_i = \mu\lambda_i$ and used μ to control the algorithm. Polyak (1992) used $s_i = \mu$ and established convergence under the assumption that the Jacobian is full rank and second-order sufficiency and strict complementarity hold. He and Jensen (Polyak, 1992 and Jensen and Polyak, 1993) were able to prove stronger results for linear, quadratic and convex programs. They use λ_i to control the algorithm asymptotically. In Conn *et al.* (1992a) we use $s_i = \mu\lambda_i^{\alpha}$, where $0 < \alpha \leq 1$, with multiplier updates given by (5.13) when appropriate. We accept or reject the multiplier update after approximate inner minimization based upon the relative degree to which we satisfy the complementary slackness conditions written as $c_i\lambda_i^{+}/s_i$. If the multiplier update is rejected then we update the penalty parameter. We include a complete convergence analysis and prove that the penalty parameter is updated only a finite number of times. In addition, asymptotically we require only one inner iteration per outer iteration (see Conn *et al.*, 1992e). Finally, we shift the starting point via an auxiliary problem when necessary (see Conn *et al.*, 1992a, for details)

Now let us consider the numerical results for this Lagrangian barrier approach — more precisely, we consider the modified barrier approach of Jensen *et al.* (1992) with additional quadratic terms. For the discrete plate problem above, it now takes 31 minutes and 54 seconds to determine the solution, which is clearly much better than running LANCELOTwith the default options.[9]

However, to emphasize some of the difficulties inherent in evaluating software for large-scale problems, when we tried different values of the penalty parameter within LANCELOT A (the results obtained with Jensen *et al.*, 1992, already included some *tuning*) we obtained the result in 4 hours, 24 minutes and 28 seconds, which already represents considerable improvement over the time using the default penalty parameter value. This improvement is especially noteworthy when one considers that LANCELOT solves the problem in 9,301 variables as opposed to the 343 of the barrier approach. For the record, MINOS 5.3 took 2 hours, 36 minutes and 30 seconds and MINOS 5.4 took 1 hour and 30 minutes.

But the story is not yet over. With a little more thought one can rearrange the linear algebra so that the *effective size* of the augmented Lagrangian approach is equivalent to that of the Lagrangian barrier approach.

To see this, consider Newton's method for minimizing the augmented Lagrangian, with slacks y added to the inequalites. Then the corresponding aug-

[9] Apparently Ben-Tal and Bendsøe (1993) is an even more successful approach to this class of structural problems.

mented Lagrangian becomes

$$\Phi(x, y, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i (c_i(x) - y_i) + \frac{1}{2\mu} \sum_{i=1}^{m} (c_i(x) - y_i)^2, \qquad (5.16)$$

The linear system that arises is given by

$$\begin{pmatrix} B + \frac{1}{\mu} A^T A & -A^T/\mu \\ -A/\mu & I/\mu \end{pmatrix} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = - \begin{pmatrix} g + A^T \bar{\lambda} \\ -\bar{\lambda} \end{pmatrix}. \qquad (5.17)$$

Noting that the coefficient matrix can be factored as

$$\begin{pmatrix} I & -A^T/\mu \\ 0 & I/\mu \end{pmatrix} \begin{pmatrix} B & 0 \\ -A & I \end{pmatrix} \qquad (5.18)$$

one can determine the search direction from[10]

$$\begin{array}{rcl} q_y & = & -\mu \bar{\lambda} \\ q_x & = & -(g + A^T \bar{\lambda}) + \frac{1}{\mu} A^T q_y \\ B p_x & = & q_x \\ \text{and } p_y & = & q_y - A p_x. \end{array} \qquad (5.19)$$

Clearly the only significant work is the third equation with the coefficient matrix B. Details are given in Conn *et al.* (1992h).

At this point it is worth mentioning that eliminating slack variables is not the only motivation for considering Lagrangian barrier techniques. In particular, the success of interior methods in linear programming (see, for example, Shanno, 1994, in this volume) suggests that they may be less sensitive to degeneracy. Moreover there is numerical evidence that the Lagrangian barrier approach is superior to standard barrier function methods when applied to problems with simple bounds (see Conn *et al.*, 1993g and Nash *et al.*, 1993) and preliminary evidence suggests that the same is true for more general constraints (see Breitfeld and Shanno, 1993a, and Breitfeld and Shanno, 1993b).

However, there are difficulties associated with the fact that one needs to remain feasible with respect to the *shifted* constraints, the fact that we lack experience (even for small dense problems) with this approach and finally and perhaps most importantly, the fact that a quadratic model is not a good model for a logarithmic barrier function.

In our attempts to improve the current version of LANCELOT we have continued our research along both theoretical and practical lines. One area we have pursued is that of using a *structured* trust region, which we motivate here by considering the following example:

$$\underset{x \in \Re^3}{\text{minimize}} \quad x_1^2 + (x_1 + x_2)^2 + e^{(x_2 + x_3)^2}. \qquad (5.20)$$

[10] Equation (5.19) is only true if none of the slacks are active, but similar simplifications are possible when there are active slacks.

Suppose we take three element functions x_1^2, $(x_1 + x_2)^2$ and $e^{(x_2+x_3)^2}$. Traditional trust region methods will tend to keep the radius unnecessarily small because of the third element, even though the first two elements are perfectly modeled by a quadratic. Thus if x_1 is far from its optimal value, it may be prevented from changing rapidly only because of a global trust region dictated by the third element. It is natural to think of using separate trust regions for *separable problems.* The idea is to generalize this by having a separate trust region for each element. In addition, we need an *overall* model on an *overall* trust region. The trust region for each element constrains only the variables for that element. Details are given in Conn *et al.* (1992f).

Another problem that one might associate with that of group partial separability is determining a suitable partitioning into groups and elements. In general this is a difficult problem to do optimally but there are two simpler versions that we have considered. The first is that of 'blowing up the internal variables' and the second is that of 'merging elements and trivial groups'. Since the main computational cost within the conjugate gradient algorithm is the multiplication of the involved matrix with a vector, we see that the cost is certainly dependent upon the representation of the matrix. In the two cases above the trade-off between computing speed and storage requirements is readily determined and can be motivated by geometrical considerations.

For element merging, say between two elements, one needs to consider the amount of overlap of the element Hessians (see Figure 1). If the area of the overlap box in the center is greater than the sum of the two areas of the zero blocks then it is worth doing merging. Details are given in Conn *et al.* (1993d).

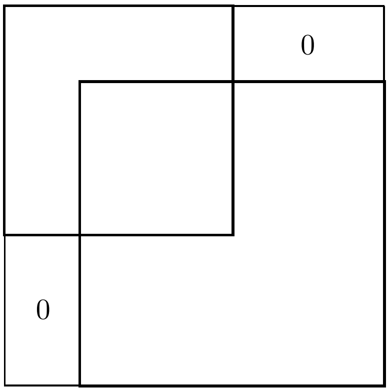

Figure 1: Two elements in the Hessian matrix

For blow up consider the following representation (Figure 2) of the blown up Hessian, the block on the left hand side, to that of its internal form, the middle block on the right hand side[11]. In this case the blow up is recommended when the total area of the last three blocks is greater than the area of the first block. Once again the reader is referred to Conn *et al.* (1993d) for details, but

[11] The last block represents the transformation matrix from elemental to internal variables and the first block on the right hand side is just its transpose.

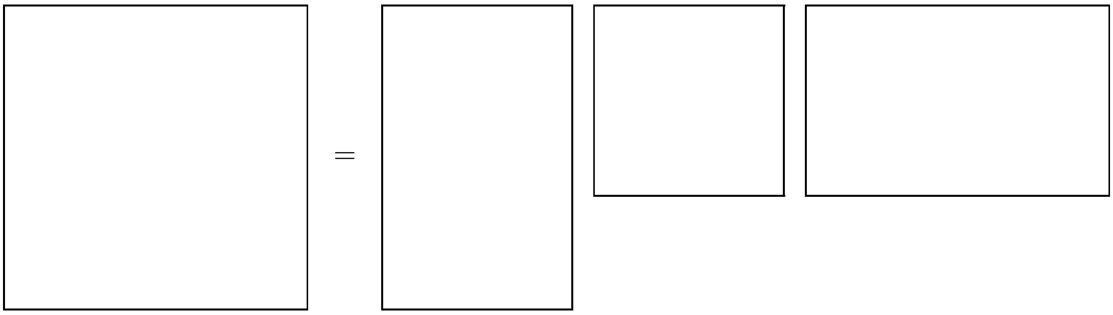

Figure 2: Elemental vs internal Hessian representation

suffice it to add that in some cases *substantial* improvements can be realized in this simple manner for both types of reformulation (see also Dayde *et al.*, 1994, for an application of similar ideas to preconditioners. In their case it often pays to merge elements even when more zeros are introduced than described here, as better vectorization is possible with bigger blocks.).

Another development for large-scale problems is our work on what we call iterated subspace minimization, which we describe below in the context of unconstrained optimization (1.1). The usual framework is to model f and then do a reasonable amount of work to 'optimize' the model, considering only what happens to the true function *one-dimensionally.* In line search methods the significant work determines the search direction and the true function determines the step-size, whilst in the case of trust region algorithms, the model determines the Cauchy point (and better) and the true function determines the trust region radius. Our motivation is that one does not appear to be using the multidimensional information obtained in the model optimization as well as one might. In addition, we observe that there are high quality algorithms available for solving small-scale problems.

This suggests the following scheme:

1. Determine a full-rank subspace matrix $S_k \in \Re^{n \times s_k}$, where $s_k \ll n$.

2. Approximately solve the s_k-dimensional minimization problem

$$\underset{y \in \Re^{s_k}}{\text{minimize}} \quad f(x_k + S_k y), \tag{5.21}$$

and set

$$x_{k+1} = \ \text{(approx)} \underset{y \in \Re^{s_k}}{\text{arg min}} \quad f(x_k + S_k y), \tag{5.22}$$

where we note that we are using the true function f in (5.22).

This begs the following important questions:

- What is a good choice for s_k?

- How do we determine the matrix S_k?
- What do we mean by "approximate" when solving problem (5.21)?
- Are there methods which are particularly appropriate for solving (5.21)?
- What can we say about the convergence of such a method?
- If we can establish convergence, what can we say about its asymptotic rate?

As we discussed at the beginning of this paper, as long as S_k contains something like the steepest descent direction with a sufficient decrease condition, global convergence is assured. Furthermore, if a Newton-like direction is also represented, we can expect a good asymptotic rate of convergence. Thus we propose to take for our columns of S_k a few directions generated by a preconditioned conjugate gradient algorithm, including the first, and a truncated Newton direction.

As an indication of the usefulness of CUTE, we were able to readily test this idea on thirty-nine unconstrained problems in the database[12]. The average size of the problems tested was around one thousand variables. Compared with the default version of LANCELOT, the new idea was at least twice as fast eleven times, at least ten times as fast twice and twice as slow five times. The remaining problems had comparable times. Details are given in Conn *et al.* (1994).

In many ways LANCELOT A's major defect is in the way it handles linear constraints. Incorporating them into an augmented Lagrangian function increases their complexity. Thus, in addition to keeping simple bounds explicitly outside the objective function, we wanted to also consider doing the same for linear constraints. The difficulty is that although it is trivial to carry out projections to maintain feasibility with respect to the bounds, it is not so trivial to do the same for linear constraints. In an attempt to improve on this, we first looked at a more general approach that made use of *inexact* projections on convex constraints. We used an approximate generalized Cauchy point and required that Goldstein-like conditions are met. Briefly, we require a feasible step within the trust region, sufficient decrease on the model functions, a sufficiently large step to prevent premature termination and we ask that we do a fixed percentage as well as the minimum value of the linearized model in the intersection of the feasible region within a ball of radius greater than or equal to the step. Details are given in Conn *et al.* (1993b). In the case of nonlinear networks, Sartenaer (1993) has obtained some very encouraging numerical results along these lines.

In addition we have extended our previous theory developed for the augmented Lagrangian function to the case where the linear constraints are not incorporated into the objective function. Moreover, as for Karmarkar (1984), we do not exclude the possibility of incorporating the simple bounds into the

[12] That is all those available, with the exception of problems which took excessive CPU time (more than 30 minutes) or were variations on the reported problems.

objective function. The inner iterations are terminated when we are 'sufficiently critical' — based upon identification of the *linear* constraints that are 'dominant'[13]. Details are given in Conn *et al.* (1993e) and Conn *et al.* (1993f) and we are currently preparing similar results for the Lagrangian barrier and mixed cases. We also incorporate the possibility of partitioning the constraints, with separate penalty parameters associated with each partition.

It should be pointed out that these issues are also relevant for the case of near-linear constraints, where, in particular, the idea used in MINOS of considering the deviation from linearity should be a good one.

Some work has been carried out to exploit particular computer architectures. The assumed partially separable form may be exploited in many ways on parallel machines (see, for example, Saludjian, 1993, and Dayde *et al.*, 1994). British Gas are currently experimenting with a parallel LANCELOT-like method for the national gas network.

6 Other Recent Progress

Let us now look at some of the recent work of our colleagues. We first consider the trust region approach of Lalee *et al.* (1993) that is designed for equality constrained problems. The method uses either exact second derivatives or limited memory quasi-Newton. It is intended for large-scale problems and is based on the trust region approach of Omojokun (1991). It uses two trust region problems:

- a *vertical step* that determines the nearest feasible point (measured by the norm of the linearized residuals) in a shrunken trust region, and
- a *horizontal step* that minimizes the model function in the trust region restricted to the null space of the constraint gradients.

This has been implemented as the algorithm ETR and a suitable interface using the CUTE tools has been written.

Defining Z as in reduced gradient methods, B as an approximation to the Hessian of the Lagrangian, the subscript k to denote iteration k, and

$$A_k^T = \begin{pmatrix} B & N \end{pmatrix} \tag{6.1}$$

$$Z^T = -\begin{pmatrix} B^{-1}N & I \end{pmatrix}, \tag{6.2}$$

the vertical step is given by

$$\underset{v \in \Re^n}{\text{minimize}} \quad \|A_k^T v + c_k\| \tag{6.3}$$

subject to

$$\|v\| \le 0.8\Delta_k. \tag{6.4}$$

[13] $a_i^T x - b_i \le \kappa_1 \omega$, for some constant κ_1, where ω is the stationarity tolerance.

Writing $d = v_k + Z_k u$, the horizontal step is given by

$$\underset{u \in \Re^n}{\text{minimize}} \quad (g_k + B_k v_k)^T Z_k u + \frac{1}{2} u^T Z_k^T B_k Z_k u \tag{6.5}$$

subject to

$$\|Z_k u\| \le \Delta_k. \tag{6.6}$$

The merit function is $f(x) + \mu\|c(x)\|_2$. The implementation uses a modification of MA28 (Duff, 1977) and the limited memory method uses a new compact representation (Byrd *et al.*, 1994), since otherwise the matrix-vector products do not take advantage of sparsity and must work with the Hessian rather than its inverse.

Motivated by real problems in process engineering, Biegler *et al.* (1993b) and Biegler *et al.* (1993a) have an implementation of the algorithm of Coleman and Conn (1982) and Coleman and Conn (1984) that is a quasi-Newton algorithm. It is designed for large-scale problems with a small associated reduced space. The reduced Hessian is updated but a correction vector is incorporated to approximate the cross term $Z^T BY d_y$, where Z is the matrix whose columns span the null space of the activity gradients, Y does the same for the range space and d_y is the component of the step in the range space. This is done with little extra cost and a one-step Q superlinear asymptotic convergence rate is maintained. The fundamental issue is that, for large-scale problems, computing orthogonal bases is expensive. However, with non-orthogonal bases, the vertical or range space component $Y d_y$ can be very large and ignoring the cross term can result in a poor step. Thus Biegler *et al.* use updates on $Z^T B$ and then compute $Z^T B(Y d_y)$ and $Z^{+^T} B^+(Y d_y)$. The former is used in the horizontal or null space step and the latter is used to update $Z^T BZ$. Moreover, these steps are 'juggled' differently, the first being used to guarantee sufficient descent and the second to ensure boundedness of $Z^T BZ$. An interface for CUTE is available. The approach has been extended via limited memory quasi-Newton to the case where the reduced space is not small, again making use of the compact representation of Byrd *et al.* (1994).

An extension of generalized reduced gradient methods (Abadie and Carpentier, 1966 and Lasdon *et al.*, 1978) to the large-scale case has been incorporated in CONOPT (Drud, 1985, Drud, 1993) and in the work of Smith and Lasdon (1992), which also makes use of the limited memory approach.

A unique approach is that of reflective Newton methods (Coleman and Li, 1992d and Coleman and Li, 1992c). This relates to the idea of replacing $x_i \ge 0$ by $x_i = |y_i|$ and replacing

$$\underset{x \in \Re^n}{\text{minimize}} \quad f(x) \tag{6.7}$$

subject to

$$x \ge 0 \tag{6.8}$$

by

$$\underset{y \in \Re^n}{\text{minimize}} \quad \hat{f}(y), \tag{6.9}$$

where $\hat{f}(y) = f(|x|)$. Amongst its advantages is the fact that this transformation does not introduce new minima, one is able to use fixed data structures and strict feasibility is maintained. The method is designed for large problems. The piecewise linear path in x-space that corresponds to a search direction in y is easily determined. They need a generalization of the Armijo-Goldstein criteria and a condition to ensure constraint compatibility. This latter definition ensures that if x is close to a boundary one is able to take a large enough step[14]. In addition, a consistency property guarantees that a first-order step converging to zero implies convergence to a stationary point. The method is implemented for simple bounds and is currently being extended to linear equality constraints. The Newton-like iterations are carried out in a scaled trust-region framework, solved in a dog-leg like method (see Powell, 1975 and Munksgaard and Reid, 1983). Asymptotically a step-size of one is taken and thus second-order convergence is attained.

We have already mentioned that barrier/interior techniques are currently a very active area of research. Nash and Sofer (1993) use a logarithmic barrier function and handle the associated ill-conditioning by using an approximate (explicit) formula for the Newton direction. This formulation (that projects orthogonally to the constraints that cause the ill-conditioning) becomes more accurate as the penalty parameter becomes smaller. In addition they use a special line search as in Murray and Wright (1976) (see also Murray and Wright, 1992), a preconditioned truncated Newton method and extrapolation as in Fiacco and McCormick (1968). Finally they use an expanded form of the Hessian of the barrier function and finite differences to derive efficient matrix-vector products. They report numerical results on simple bound constrained problems as large as 100,000 variables. Nash *et al.* (1993) use a similar implementation of a modified (shifted/Lagrangian) logarithmic barrier function with additional quadratic terms. More specifically, writing $t = c_i(x)$ and considering a single barrier term Ψ, they use the term

$$\begin{aligned} \Psi(\mu^{-1}t + 1) \quad &= \log(\mu^{-1}t + 1) \quad && \text{if } t \geq -\mu/2, \\ &= q(t) && \text{if } t < -\mu/2. \end{aligned} \tag{6.10}$$

Here the quadratic, q, interpolates $q(-\mu/2), q^{'}(-\mu/2)$ and $q^{''}(-\mu/2)$ with the corresponding logarithmic values. Interestingly, in this context they abandoned the special line search of Murray and Wright (1976)[15]. The numerical results reported were better than using just the barrier function.

Breitfeld and Shanno had similar computational results. They used CONMIN (Shanno and Phua, 1980), which is a limited memory BFGS/CG algorithm. In Breitfeld and Shanno (1993a), they suggested replacing equalities by

[14] This, in turn, ensures that the distance to breakpoints remains bounded away from zero. Reflections are likely to be suitable if the angle is reasonable.

[15] This suggests that the singularity can be taken care of by adding quadratic terms rather than using a special line search. Breitfeld and Shanno (1993b) made the same observation.

two inequalities which are then shifted. They claim that this is preferable to using an augmented Lagrangian to handle equalities. However, the numerical results to date must still be considered very preliminary.

We now report on some numerical experience and testing in general. Extensive numerical results are available for LANCELOT in Conn *et al.* (1992c) and Conn *et al.* (1993c). These describe tests using all the LANCELOT options on about one thousand problem instances. The basic conclusions are that LANCELOT appears to be very robust and the symmetric rank one update is the best quasi-Newton update in that trust-region context (see also Byrd *et al.*, 1993a, who based upon their convergence analysis, recommend updating even when steps are rejected). ¿From the point of view of general comparisons, there is not a great deal of large-scale experience[16] in the published literature. Eldersveld *et al.* (1993) looked at very sparse problems that have the possibility of having a large reduced space (dimension greater than 700) and where the functions are expensive to evaluate. They considered 109 problems with from 40 to 2,400 variables. They compared NPSOL (Gill *et al.*, 1986), which was not designed for large sparse problems; MINOS, which, although designed for the large-scale case, prefers small reduced subspaces; NLPSPR (Betts and Frank, 1994), which is a sequential quadratic programming method that uses Schur complements on an (increasing) Karush-Kuhn-Tucker matrix; and LSSQP (Gill *et al.*, 1993a and Gill *et al.*, 1993b), which is a transformed Hessian method. Their main conclusions were that NLPSPR was best (although they admit a bias since the code was designed for the class of problems they tested), MINOS was rather disappointing, NPSOL was robust for those problems for which enough storage was available and LSSQP performs well when the reduced space is less than two hundred dimensional. We are currently (Bongartz *et al.*, 1994c), doing an extensive comparison between MINOS and LANCELOT using the CUTE database. We would like to identify, amongst other details, the class of problems for which each is most appropriate and verify if these findings agree with our preconceptions. As for Eldersveld *et al.* (1993), our preliminary results are that MINOS is not as robust as one would hope, but one should bear in mind that, firstly, we have more expertise with LANCELOT and, secondly, the basis for MINOS is now rather old technology.

Bouaricha and colleagues (Bouaricha and Gould, 1994, Bouaricha and Schnabel, 1994a, Bouaricha and Schnabel, 1994b and Bouaricha and Tuminaro, 1994) are extending the earlier work on tensor methods of Schnabel and Chow (1991) and Schnabel and Frank (1984) to large-scale problems. The basic idea of tensor methods is to base each iteration on a higher order model than standard methods, but in such a way that there is almost no increased cost. The motivation is to improve upon the standard methods when applied to non-singular and (especially) singular problems. As for non-tensor methods, the extension to large problems suggests the use of finite differences, the replacement of orthogonal decompositions and the exploitation of structure. However, because of the nature of the tensor terms, a basic question here is will information in small dimensional subspaces (and in what form) help when the underlying problem

[16] Indeed, there isn't much recent extensive testing for small-scale problems.

is large?

Finally, we should not forget there are methods designed for specially structured large-scale nonlinear programming problems. Some examples follow.

- For nonlinear least-squares problems: Ben Daya and Shetty (1988), Coleman and Plassman (1988), Coleman and Plassman (1992), Golub *et al.* (1986), Gulliksson (1990), Gulliksson (1993), Kaufman and Sylvester (1993), Toint (1987b) and Toint (1987a).

- For minimax, l_p, l_2 and l_∞ problems: Coleman and Li (1992a), Coleman and Li (1992b), Dax (1993), Li (1993b), Li (1993a), Jónasson and Madsen (1992) and Sklar and Armstrong (1993).

- For quadratic programming problems (including those constrained only by simple bounds): Coleman and Hulbert (1993a), Coleman and Hulbert (1993b), Gould (1991), Júdice and Pires (1989), Moré and Toraldo (1989), Moré and Toraldo (1991), Soares *et al.* (1993) and Vanderbei and Carpenter (1993).

- For nonlinear network problems: Ahlfeld *et al.* (1987), Dembo (1986), Sartenaer (1993), Toint and Tuyttens (1990), Toint and Tuyttens (1992), Zenios and Mulvey (1986), Zenios and Mulvey (1988), Zenios and Pinar (1989).

- For location problems: Bongartz *et al.* (1994a), Bongartz *et al.* (1994b) and Calamai and Conn (1987).

- For linear complementarity problems: Júdice and Pires (1993) — see also Júdice (1994), in this volume.

.

Finally we have said little about automatic differentiation or special architectures. The former still do not seem to have had as much impact in optimization as one might have hoped. Besides the chapter in this volume (Dixon, 1994), we refer the reader to Bischof *et al.* (1991), Bischof and Griewank (1992), Dixon *et al.* (1988), Griewank (1989), Griewank and Corliss (1991) and Griewank *et al.* (1993). For the latter the reader is referred to the chapter of Schnabel in this volume (Schnabel, 1994) and Zenios (1989).

7 In conclusion

We hope we have convinced some of you that it is possible to solve large nonlinear problems in thousands of variables in acceptable time on reasonable workstations. Moreover software packages are available and it is worth pointing out that, although they are designed for large-scale problems, some of them can nevertheless be excellent for the small-scale case. Our hope is that, in the not too distant future, practitioners will be solving nonlinear models rather than linear ones, when the former is the most appropriate one to consider. We also have taken some pains to emphasize the importance of testing. In our opinion, nobody should be publishing papers whose main purpose is to describe an algorithm that is intended to be practically useful, unless they also provide evidence that the algorithm is competitive on significant problems. Even more obvious is the statement that it is meaningless to propose algorithms for large-scale problems and report numerical results only for problems in a few hundred variables.

Besides the relevant chapters in this volume, very good background reading in linear, constrained and unconstrained nonlinear programming is provided in the chapters of Goldfarb and Todd (1989), Dennis and Schnabel (1989) and Gill and Murray (1989) in the book by Nemhauser *et al.* (1989). Recent articles and books devoted primarily to large-scale optimization include Coleman and Li (1990), Coleman (1993), Conn *et al.* (1989), Conn *et al.* (1990b), Conn *et al.* (1992b), Conn *et al.* (1992g) and Wright (1991). The book by Moré and Wright (1993), besides having a useful introduction to the theory, indicates the available software. Some examples of applications are given in Biegler (1992), Chinchalkar and Coleman (1993), Coleman and Liao (1993), Coleman *et al.* (1992), Dunn (1993), Falk and McCormick (1986), Hager (1990), Jones (1967), Kunish and Sachs (1992), Liao (1993), McCormick (1972), McCormick and Sofer (1991), Schrady and Choe (1971), Werbos (1988) and Wu (1993).

Finally, in a subject this complex, a single short article, necessarily, is only able to give an idea of the nature of the main issues in the current research. Moreover we have no doubt that our own particular biases show. Nevertheless we hope that the text and the references will be useful to those interested in what currently is an exciting and vibrant research area.

References

[Abadie and Carpentier, 1966] J. Abadie and J. Carpentier. Généralisation de la méthode du gradient réduit de Wolfe au cas des contraintes non-linéaires. In D.B. Hertz and J. Melese, editors, *Proceedings IFORS Conference*, pages 1041–1053. J. Wiley and Sons, Amsterdam, 1966.

[Achtziger *et al.*, 1992] M. Achtziger, M. P. Bendsøe, A. Ben-Tal, and J. Zowe. Equivalent displacement based formulations for maximum strength topology design. *Impact of Computing in Science and Engineering*, 4:315–345, 1992.

[Ahlfeld *et al.*, 1987] D. P. Ahlfeld, R. S. Dembo, J. M. Mulvey, and S. A. Zenios. Nonlinear programming on generalized networks. *ACM Transactions on Mathematical Software*, 13(3):350–367, 1987.

[Arioli *et al.*, 1993] M. Arioli, T. F. Chan, I. S. Duff, N. I. M. Gould, and J. K. Reid. Computing a search direction for large-scale linearly constrained nonlinear optimization calculations. Technical Report TR/PA/93/34, CERFACS, Toulouse, France, 1993.

[Averick and Moré, 1991] B. M. Averick and J. J. Moré. The Minpack-2 test problem collection. Technical Report ANL/MCS-TM-157, Argonne National Laboratory, Argonne, USA, 1991.

[Averick *et al.*, 1991] B. M. Averick, R. G. Carter, and J. J. Moré. The Minpack-2 test problem collection (preliminary version). Technical Report ANL/MCS-TM-150, Argonne National Laboratory, Argonne, USA, 1991.

[Bartholomew-Biggs and Hernandez, 1994] M. C. Bartholomew-Biggs and M. de F. G. Hernandez. Modifications to the subroutine OPALQP for dealing with large problems. *Journal of Economic Dynamics and Control*, 18:185–204, 1994.

[Bartholomew-Biggs, 1994] M. C. Bartholomew-Biggs. Algorithms for general constrained nonlinear optimization. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Ben Daya and Shetty, 1988] M. Ben Daya and C. M. Shetty. Polynomial barrier function algorithm for convex quadratic programming. Research Report J85-5, School of ISE, Georgia Institute of Technology, Atlanta, Georgia, 1988.

[Ben-Tal and Bendsøe, 1993] A. Ben-Tal and M. P. Bendsøe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3(2):322–358, 1993.

[Bertsekas, 1982] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2):221–246, 1982.

[Betts and Frank, 1994] J. T. Betts and P. D. Frank. A sparse nonlinear optimization algorithm. *Journal of Optimization Theory and Applications*, 82(3), 1994, to appear.

[Biegler *et al.*, 1993a] L.T. Biegler, J. Nocedal, and C. Schmid. Numerical experience with a reduced Hessian method for large-scale constrained optimization. Research Report (in preparation), EE and CS, Northwestern University, Evanston, Illinois, 1993.

[Biegler *et al.*, 1993b] L.T. Biegler, J. Nocedal, and C. Schmid. A reduced Hessian method for large-scale constrained optimization. Research Report NAM-03, EE and CS, Northwestern University, Evanston, Illinois, 1993.

[Biegler, 1992] L.T. Biegler. Optimization strategies for complex process models. *Advances in Chemical Engineering*, 18:197–256, 1992.

[Bischof and Griewank, 1992] C. Bischof and A. O. Griewank. ADIFOR: A FORTRAN system for portable automatic differentiation. Technical Report MCS-P317-0792, Argonne National Laboratory, Argonne, USA, 1992.

[Bischof *et al.*, 1991] C. Bischof, A. Carle, G. Corliss, P. Hovland, and A. O. Griewank. ADIFOR: Generating derivative codes from Fortran programs. Technical Report MCS-P263-0991, Argonne National Laboratory, Argonne, USA, 1991.

[Björck, 1994] A. Björck. Generalized and sparse least squares problems. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Boggs and Tolle, 1989] P. T. Boggs and J. W. Tolle. A strategy for global convergence in a sequential quadratic programming algorithm. *SIAM Journal on Numerical Analysis*, 26(3):600–623, 1989.

[Bongartz *et al.*, 1993] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and Unconstrained Testing Environment. Research Report RC 18860, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Bongartz *et al.*, 1994a] I. Bongartz, P. H. Calamai, and A. R. Conn. A projection method for l_p norm location-allocation problems. *Mathematical Programming*, 1994, to appear.

[Bongartz *et al.*, 1994b] I. Bongartz, P. H. Calamai, and A. R. Conn. A second-order algorithm for the continuous capacitated location-allocation problem. Research Report (in preparation), IBM T. J. Watson Research Center, Yorktown Heights, USA, 1994.

[Bongartz *et al.*, 1994c] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A numerical comparison between the LANCELOT and MINOS packages for large-scale nonlinear optimization. Research Report (in preparation), IBM T. J. Watson Research Center, Yorktown Heights, USA, 1994.

[Bouaricha and Gould, 1994] Ali Bouaricha and N. I. M. Gould. Tensor methods for large sparse unconstrained minimization problems. Technical report (in preparation), CERFACS, Toulouse, France, 1994.

[Bouaricha and Schnabel, 1994a] Ali Bouaricha and R.B. Schnabel. A software package for large sparse nonlinear equations using tensor methods. Technical report, (in preparation), CERFACS, Toulouse, France, 1994.

[Bouaricha and Schnabel, 1994b] Ali Bouaricha and R.B. Schnabel. A software package for large sparse nonlinear least squares using tensor methods. Technical report, (in preparation), CERFACS, Toulouse, France, 1994.

[Bouaricha and Tuminaro, 1994] Ali Bouaricha and R. Tuminaro. Tensor-Krylov methods for large nonlinear equations on sequential and parallel computers. Technical report, (in preparation), CERFACS, Toulouse, France, 1994.

[Breitfeld and Shanno, 1993a] M. G. Breitfeld and D. Shanno. Preliminary computational experience with modified log-barrier functions for large-scale nonlinear programming. Research Report RRR 08-93, Rutgers Center for Operations Research, New Brunswick, USA, 1993.

[Breitfeld and Shanno, 1993b] M. G. Breitfeld and D. F. Shanno. Computational experience with modified log-barrier methods for nonlinear programming. Research Report RRR 17-93, Rutgers Center for Operations Research, New Brunswick, USA, 1993.

[Buckley, 1989] A. G. Buckley. Test functions for unconstrained minimization. Technical Report CS-3, Computing Science Division, Dalhousie University, Halifax, Canada, 1989.

[Byrd *et al.*, 1993a] R. H. Byrd, H.F. Khalfan, and R. B. Schnabel. Analysis of a symmetric rank-one trust region method. Technical Report CU-CS-657-93, Department of Computer Science, University of Colorado at Boulder, Boulder, USA, 1993.

[Byrd *et al.*, 1993] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. Technical Report NAM-08, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois, 1993.

[Byrd *et al.*, 1994] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representation of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming, Series A*, 1994, to appear.

[Calamai and Conn, 1987] P. H. Calamai and A. R. Conn. A projected newton method for l_p norm location problems. *Mathematical Programming*, 38:75–109, 1987.

[Chinchalkar and Coleman, 1993] S. Chinchalkar and T. F. Coleman. Parallel structural optimization applied to bone remodeling on distributed memory machines. Technical Report CTC93TR146, Advanced Computing Research Institute, Cornell Theory Center, Cornell University, Ithaca, USA, 1993.

[Coleman and Conn, 1982] T. F. Coleman and A. R. Conn. Nonlinear programming via an exact penalty function method: Global analysis. *Mathematical Programming*, 24(3):137–161, 1982.

[Coleman and Conn, 1984] T. F. Coleman and A. R. Conn. On the local convergence of a quasi-Newton method for the nonlinear programming problem. *SIAM Journal on Numerical Analysis*, 21(4):755–769, 1984.

[Coleman and Hulbert, 1993a] T. F. Coleman and L. Hulbert. A direct active set method for large sparse quadratic programs with simple bounds. *Mathematical Programming*, pages 373–406, 1993.

[Coleman and Hulbert, 1993b] T. F. Coleman and L. Hulbert. A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds. *SIAM Journal on Optimization*, 3:298–321, 1993.

[Coleman and Li, 1990] T. F. Coleman and Y. Li, editors. *Large Scale Numerical Optimization.* SIAM, Philadelphia, USA, 1990.

[Coleman and Li, 1992a] T. F. Coleman and Y. Li. A global and quadratic affine scaling method for linear l_1 problems. *Mathematical Programming*, 56:189–222, 1992.

[Coleman and Li, 1992b] T. F. Coleman and Y. Li. A global and quadratically-convergent method for linear l_∞ problems. *SIAM Journal on Scientific and Statistical Computing*, 29:1166–1186, 1992.

[Coleman and Li, 1992c] T. F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. Technical Report CTC 92TR110, Cornell Theory Center, Ithaca, USA, 1992.

[Coleman and Li, 1992d] T. F. Coleman and Y. Li. A reflective Newton method for minimizing a quadratic function subject to bounds on the variables. Technical Report CTC 92TR111, Cornell Theory Center, Ithaca, USA, 1992.

[Coleman and Liao, 1993] T. F. Coleman and A. Liao. An efficient trust region method for unconstrained discrete-time optimal control problems. Technical Report CTC93TR144, Advanced Computing Research Institute, Cornell Theory Center, Cornell University, Ithaca, USA, 1993.

[Coleman and Plassman, 1988] T. F. Coleman and P. E. Plassman. Solution of nonlinear least-squares problems on a multiprocessor. In *Parallel Computing*, pages 44–80. Springer Verlag, Berlin, 1988.

[Coleman and Plassman, 1992] T. F. Coleman and P. E. Plassman. A parallel nonlinear least-squares solver: theoretical analysis and numerical results. *SIAM Journal on Scientific and Statistical Computing*, 13:771–793, 1992.

[Coleman *et al.*, 1992] T. F. Coleman, D. Shalloway, and Z. Wu. Isotropic effective energy simulated annealing searches for low energy molecular cluster states. Technical Report CTC92TR113, Advanced Computing Research Institute, Cornell Theory Center, Cornell University, Ithaca, USA, 1992.

[Coleman, 1993] T. F. Coleman. Large Scale Numerical Optimization: Introduction and overview. In J. Williams and A. Kent, editors, *Encyclopedia of Computer Science and Technology*, Volume 28, supplement 13, pages 167–196. Marcel Dekker, New York, USA, 1993.

[Conn *et al.*, 1988a] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25:433–460, 1988. See also same journal, 26:764–767, 1989.

[Conn *et al.*, 1988b] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation*, 50:399–430, 1988.

[Conn *et al.*, 1989] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Large-scale optimization. *Mathematical Programming, Series B*, 45(3), 1989.

[Conn *et al.*, 1990a] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project. In R. Glowinski and A. Lichnewsky, editors, *Computing Methods in Applied Sciences and Engineering*, pages 42–54. SIAM, Philadelphia, USA, 1990.

[Conn *et al.*, 1990b] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Large-scale optimization — applications. *Mathematical Programming, Series B*, 48(1), 1990.

[Conn *et al.*, 1991] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 28(2):545–572, 1991.

[Conn *et al.*, 1992a] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. Research Report RC 18049, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Conn *et al.*, 1992b] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*, Volume 17 of *Springer Series in Computational Mathematics*. Springer Verlag, Heidelberg, Berlin, New York, 1992.

[Conn *et al.*, 1992c] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization. Research Report RC 18434, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Conn *et al.*, 1992d] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear equality constraints and simple bounds. In D.F Griffiths and G.A. Watson, editors, *Proceedings of the 14th Biennial Numerical Analysis Conference Dundee 1991*, pages 49–68. Longmans, 1992.

[Conn *et al.*, 1992e] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear inequality constraints and simple bounds. Research Report RC 18382, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Conn *et al.*, 1992f] A. R. Conn, Nick Gould, and Ph. L. Toint. Convergence properties of minimization algorithms for convex constraints using a structured trust region. Research Report RC 18274, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Conn *et al.*, 1992g] A. R. Conn, Nick Gould, and Ph. L. Toint. Large-scale nonlinear constrained optimization. In Jr. R. E. O'Malley, editor, *Proceedings of the Second International Conference on Industrial and Applied Mathematics*, pages 51–70. SIAM, Phildelphia, USA, 1992. (Also in M.S. Moonen, G.H. Golub and B.L.R DeMoor, editors, *Linear Algebra for Large-Scale and Real-Time Applications*, Volume 232 of *NATO ASI Series E: Applied Sciences.* Kluwer Academic Publishers, 1993.)

[Conn *et al.*, 1992h] A. R. Conn, Nick Gould, and Ph. L. Toint. A note on exploiting structure when using slack variables. Research Report RC 18435, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Conn *et al.*, 1993a] A. R. Conn, N. I. M. Gould, M. Lescrenier, and Ph. L. Toint. Performance of a multifrontal scheme for partially separable optimization. In *Advances in numerical partial differential equations and optimization, Proceedings of the Sixth Mexico-United States Workshop.* Kluwer Academic Publishers, 1993.

[Conn *et al.*, 1993b] A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal on Optimization*, 3(1):164–221, 1993.

[Conn *et al.*, 1993c] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Complete numerical results for LANCELOT Release A. Research Report RC 18750, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Conn *et al.*, 1993d] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Improving the decomposition of partially separable functions in the context of large-scale optimization: a first approach. In W. W. Hager, D. W. Hearn, and P.M. Pardalos, editors, *Large Scale Optimization: State of the Art.* Kluwer Academic Publishers, 1993.

[Conn *et al.*, 1993e] A. R. Conn, Nick Gould, A. Sartenaer, and Ph. L. Toint. Local convergence properties of two augmented Lagrangian algorithms for optimization with a combination of general equality and linear constraints. Research Report RC18901, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Conn *et al.*, 1993f] A. R. Conn, Nick Gould, A. Sartenaer, and Ph. L. Toint. Local convergence properties of a Lagrangian barrier algorithm for optimization with a combination of general inequality and linear constraints. Research Report (in preparation), IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Conn *et al.*, 1993g] A. R. Conn, Nick Gould, and Ph. L. Toint. A note on using alternative second-order models for the subproblems arising in barrier function methods for minimization. Research Report RC18898, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Conn *et al.*, 1994] A. R. Conn, Nick Gould, A. Sartenaer, and Ph. L. Toint. On iterated-subspace minimization methods for nonlinear optimization. Research Report (in preparation), IBM T. J. Watson Research Center, Yorktown Heights, USA, 1994.

[Curtis *et al.*, 1974] A. Curtis, M. J. D. Powell, and J. Reid. On the estimation of sparse Jacobian matrices. *Journal of the Institute of Mathematics and its Applications*, 13:117–119, 1974.

[Dax, 1993] A. Dax. A row relaxation method for large minmax problems. *BIT*, 33:262–273, 1993.

[Dayde *et al.*, 1994] M. Dayde, J.-Y. L'Excellent, and N. I. M. Gould. On the use of element by element preconditioners to solve large-scale partially separable optimization problems. Technical Report (in preparation), CERFACS, Toulouse, France, 1994.

[Dembo and Tulowitski, 1983] R. S. Dembo and U. Tulowitski. On the minimization of quadratic functions subject to box constraints. School of Organization and Management Working paper series B no. 71, Yale University, 1983.

[Dembo, 1984] R. S. Dembo. A primal truncated-Newton algorithm with application to large-scale nonlinear network optimization. Technical Report 72, Yale School of Management, Yale University, New Haven, USA, 1984.

[Dembo, 1986] R. S. Dembo. The performance of NLPNET, a large scale nonlinear network optimizer. *Mathematical Programming, Series B*, 26:245–249, 1986.

[Dennis and Schnabel, 1983] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, Englewood Cliffs, USA, 1983.

[Dennis and Schnabel, 1989] J. E. Dennis and R. B. Schnabel. Unconstrained optimization. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Optimization*, Volume 1 of *Handbooks in Operations Research and Management Science*, pages 73–170. North-Holland, Amsterdam, The Netherlands, 1989.

[Dixon *et al.*, 1988] L. C. W. Dixon, P. Dolan, and R. Price. Finite element optimization: the use of structured automatic differentiation. In A. Osiadacz, editor, *Simulation and Optimization of Large Systems*, pages 117–141. Oxford University Press, Oxford, 1988.

[Dixon, 1994] L. C. W. Dixon. Automatic differentiation and continuous optimization. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Drud, 1985] A. Drud. CONOPT: a GRG code for large sparse dynamic nonlinear optimization problems. *Mathematical Programming*, 31(2):153–191, 1985.

[Drud, 1993] A. Drud. CONOPT: a large-scale GRG code. Technical report, ARKI Consulting and Developing, Bagsvaerd, Denmark, 1993.

[Duff and Reid, 1982] I. S. Duff and J. K. Reid. MA27: A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Report R-10533, AERE Harwell Laboratory, Harwell, UK, 1982.

[Duff and Reid, 1983] I. S. Duff and J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[Duff and Reid, 1993] I. S. Duff and J. K. Reid. MA47: A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Research Report (to appear), Rutherford Appleton Laboratory, Chilton, England, 1993.

[Duff *et al.*, 1986] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices.* Clarendon Press, Oxford, UK, 1986.

[Duff *et al.*, 1988] I. S. Duff, N. I. M. Gould, M. Lescrenier, and J. K. Reid. The multifrontal method in a parallel environment. In M. G. Cox and S. J. Hammarling, editors, *Reliable Scientific Computation.* Oxford University Press, Oxford, UK, 1988.

[Duff, 1977] I. S. Duff. MA28: A set of Fortran subroutines for sparse unsymmetric linear equations. Report R-8730, AERE Harwell Laboratory, Harwell, UK, 1977.

[Dunn, 1993] J. C. Dunn. Second-order multiplier update calculations for optimal control problems and related large-scale nonlinear programs. *SIAM Journal on Optimization*, 3(3):489–502, 1993.

[Eldersveld *et al.*, 1993] S. K. Eldersveld, J. T. Betts, and W. P. Huffman. A performance comparison of nonlinear programming algorithms for large sparse problems. Presented at the Fourth Stockholm Optimization Days, Royal Institute of Technology, Stockholm, 16–17 August, 1993.

[Eldersveld, 1992] S. K. Eldersveld. Large-scale sequential quadratic programming algorithms. Technical Report SOL 92-4, Department of Operations Research, Stanford University, Stanford, USA, 1992.

[Falk and McCormick, 1986] J. E. Falk and G. P. McCormick. Computational aspects of the international coal trade model. In P.T. Harker, editor, *Spacial price equilibrium: Advances in theory, computation and application*, Volume 249 of *Lecture Notes in Economics and Mathematical Systems.* Springer Verlag, Berlin, 1986.

[Fiacco and McCormick, 1968] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* J. Wiley and Sons, New York, 1968. Reprinted as *Classics in Applied Mathematics 4*, SIAM, 1990.

[Fletcher, 1987] R. Fletcher. *Practical Methods of Optimization.* J. Wiley and Sons, Chichester, second edition, 1987.

[Fletcher, 1994] R. Fletcher. Algorithms for unconstrained optimization. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Forsgren and Murray, 1993] A. L. Forsgren and W. Murray. Newton methods for large-scale linear equality-constrained minimization. *SIAM Journal on Matrix Analysis and Applications*, 14:560–587, 1993.

[Freund, 1991] R. M. Freund. Theoretical efficiency of a shifted-barrier-function algorithm for linear programming. *Linear Algebra and Applications*, 152:19–41, 1991.

[George and Liu, 1981] A. George and J. W.-H. Liu. *Computer solution of large sparse positive definite systems.* Prentice-Hall, Englewood Cliffs, USA, 1981.

[Gill and Murray, 1974] P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, 28:311–350, 1974.

[Gill and Murray, 1989] P. E. Gill and W. Murray. Constrained optimization. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Optimization*, Volume 1 of *Handbooks in Operations Research and Management Science*, pages 73–170. North-Holland, Amsterdam, The Netherlands, 1989.

[Gill *et al.*, 1986] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's guide for NPSOL (version 4.0): A Fortran package for nonlinear programming. Technical Report SOL86-2, Department of Operations Research, Stanford University, Stanford, California 94305, USA, 1986.

[Gill *et al.*, 1988] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Shifted barrier methods for linear programming. Technical Report SOL88-9, Department of Operations Research, Stanford University, Stanford, California 94305, USA, 1988.

[Gill *et al.*, 1992] P. E. Gill, W. Murray, D. B. Poncеléon, and M. A. Saunders. Preconditioners for indefinite systems arising in optimization. *SIAM Journal on Matrix Analysis and Applications*, 13:292–311, 1992.

[Gill *et al.*, 1993a] P.E. Gill, W. Murray, and M.A. Saunders. Large-scale SQP methods and their application in trajectory optimization. In R. Bulirsch and D. Kraft, editors, *Control Applications of Optimization*, International Series of Numerical Mathematics. Birkhauser, Basel, Boston, Stuttgart, 1993.

[Gill *et al.*, 1993b] P.E. Gill, W. Murray, and M.A. Saunders. Transformed Hessian methods for large-scale constrained optimization. Presented at the Fourth Stockholm Optimization Days, Royal Institute of Technology, Stockholm, 16–17 August, 1993.

[Goldfarb and Todd, 1989] D. Goldfarb and M. J. Todd. Linear programming. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Optimization*, Volume 1 of *Handbooks in Operations Research and Management Science*, pages 73–170. North-Holland, Amsterdam, The Netherlands, 1989.

[Golub and Loan, 1989] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, second edition, 1989.

[Golub *et al.*, 1986] G. H. Golub, P. E. Manneback, and Ph. L. Toint. A comparison between some direct and iterative methods for large scale geodetic least squares problems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):799–816, 1986.

[Gould, 1991] N. I. M. Gould. An algorithm for large-scale quadratic programming. *IMA Journal of Numerical Analysis*, 11(3):299–324, 1991.

[Griewank and Corliss, 1991] A. Griewank and G. F. Corliss. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application.* SIAM, Philadelphia, USA, 1991.

[Griewank and Toint, 1982a] A. Griewank and Ph. L. Toint. On the unconstrained optimization of partially separable functions. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312. Academic Press, London and New York, 1982.

[Griewank and Toint, 1982b] A. Griewank and Ph. L. Toint. Partitioned variable metric updates for large structured optimization problems. *Numerische Mathematik*, 39:429–448, 1982.

[Griewank *et al.*, 1993] A. O. Griewank, D. Juedes, J. Srinivasan, and C. Tyner. ADOL-C, a package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, 1993, to appear.

[Griewank, 1989] A. Griewank. On automatic differentiation. In M. Iri and K. Tanabe, editors, *Mathematical Programming: recent developments and applications*, pages 83–108. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1989.

[Gulliksson, 1990] M. Gulliksson. *Algorithms for Nonlinear Least Squares with Applications to Orthogonal Regression.* PhD thesis, Institute of Information Processing, University of Umeå, S-901 87 Umeå, Sweden, 1990.

[Gulliksson, 1993] M. Gulliksson. Algorithms for weighted nonlinear least squares problems — especially surface fitting problems. Presented at the Fourth Stockholm Optimization Days, Royal Institute of Technology, Stockholm, 16–17 August, 1993.

[Hager, 1990] W. W. Hager. Multipliers methods for nonlinear optimal control. *SIAM Journal on Numerical Analysis*, 27(4):1061–1080, 1990.

[Harwell Subroutine Library, 1993] Harwell Subroutine Library. *A catalogue of subroutines (release 11).* Advanced Computing Department, Harwell Laboratory, Harwell, UK, 1993.

[Hebden, 1973] M. D. Hebden. An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, UK, 1973.

[Hock and Schittkowski, 1981] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*, Volume 187 of *Lectures Notes in Economics and Mathematical Systems.* Springer Verlag, Berlin, 1981.

[International Business Machines Corporation, 1990] International Business Machines Corporation. *Optimization Subroutine Library: Guide and Reference*, second edition, 1990.

[Jensen and Polyak, 1993] D. Jensen and R. Polyak. On the convergence of a modified barrier method for convex programming. Research Report RC 18570, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1993.

[Jensen *et al.*, 1992] D. Jensen, R. Polyak, and R. Schneur. Numerical experience with modified barrier functions method for linear programming. Research Report RC 18415, IBM T. J. Watson Research Center, Yorktown Heights, USA, 1992.

[Jittorntrum and Osborne, 1980] K. Jittorntrum and M. Osborne. A modified barrier function method with improved rate of convergence for degenerate problems. *Journal of the Australian Mathematical Society (Series B)*, 21:305–329, 1980.

[Jónasson and Madsen, 1992] K. Jónasson and K. Madsen. Corrected sequential linear programming for sparse minimax optimization. Technical Report NI-92-06, Institute for Numerical Analysis, Technical University of Denmark, 2800 Lyngby, Denmark, 1992.

[Jones, 1967] A. P. Jones. The chemical equilibrium problem: An application of SUMT. Technical Report RAC-TP-272, Research Analysis Corporation, Research Analysis Corporation, McLean, Virginia, USA, 1967.

[Júdice and Pires, 1989] J. J. Júdice and F. M. Pires. Direct methods for convex quadratic programs subject to box constraints. Technical report, Universidade de Coimbra, 300 Coimbra, Portugal, 1989.

[Júdice and Pires, 1993] J. J. Júdice and F. M. Pires. A block principal pivoting algorithm for large-scale strictly monotone linear complementary problems. Technical report, Universidade de Coimbra, 300 Coimbra, Portugal, 1993. To appear in Computers and Operations Research.

[Júdice, 1994] J. J. Júdice. Algorithms for linear complementarity problems. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Karmarkar, 1984] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[Kaufman and Sylvester, 1993] L. Kaufman and G. Sylvester. Seperable nonlinear least-squares with multiple right-hand sides. *SIAM Journal on Matrix Analysis and Applications*, 13:68–89, 1993.

[Koontz *et al.*, 1985] J.E. Koontz, R.B. Schnabel, and B.E. Weiss. A modular system of algorithms for unconstrained minimization. *ACM Transactions on Mathematical Software*, 11:419–440, 1985. Also available as Technical Report CU-CS-240-82, Department of Computer Science, University of Colorado, Boulder, CO.

[Kunish and Sachs, 1992] K. Kunish and E. W. Sachs. Reduced SQP methods for parameter identification problems. *SIAM Journal on Numerical Analysis*, 29(6):1793–1822, 1992.

[Lalee *et al.*, 1993] M. Lalee, J. Nocedal, and T. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. EE and CS Technical Report, Northwestern University, Evanston, USA, 1993.

[Lasdon *et al.*, 1978] L. S. Lasdon, A. D. Waren, A. Jain, and M. Ratner. Design and testing of a generalized reduced gradient code for nonlinear programming. *ACM Transactions on Mathematical Software*, 4:34–50, 1978.

[Lescrenier, 1991] M. Lescrenier. Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM Journal on Numerical Analysis*, 28(2):476–495, 1991.

[Levenberg, 1944] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly Journal on Applied Mathematics*, 2:164–168, 1944.

[Levitin and Polyak, 1966] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Comput. Math. and Math. Phys.*, 6:1–50, 1966.

[Li, 1993a] Y. Li. A globally convergent method for l_p problems. *SIAM Journal on Optimization*, 3:609–629, 1993.

[Li, 1993b] Y. Li. Solving l_p-norm problems and applications. Technical Report CTC93TR122, Cornell Theory Center, Cornell University, Ithaca, USA, 1993.

[Liao, 1993] Aiping Liao. Some efficient algorithms for unconstrained discrete-time optimal control problems. Technical Report CTC93TR159, Advanced Computing Research Institute, Cornell Theory Center, Cornell University, Ithaca, USA, 1993.

[Liu and Nocedal, 1988] D. C. Liu and J. Nocedal. Test results of two limited memory methods for large scale optimization. Technical Report NAM 04, Department of EE and CS, Northwestern University, Evanston, IL, 1988.

[Liu and Nocedal, 1989] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming, Series B*, 45:503–528, 1989.

[Marquardt, 1963] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.

[McCormick and Sofer, 1991] G. P. McCormick and A. Sofer. Optimization with unary functions. *Mathematical Programming*, 52:167–179, 1991.

[McCormick, 1969] G. P. McCormick. Anti-zig-zagging by bending. *Management Science*, 15(5):315–320, 1969.

[McCormick, 1972] G. P. McCormick. Computational aspects of nonlinear programming solutions to large-scale inventory problems. Technical Report Technical Memorandum Serial T-63488, Department of Operations Research, George Washington University, Washington DC 20052, 1972.

[Moré and Toraldo, 1989] J. J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 14:14–21, 1989.

[Moré and Toraldo, 1991] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

[Moré and Wright, 1993] J. J. Moré and S. J. Wright. *Optimization Software Guide*. SIAM, Philadelphia, USA, 1993.

[Moré *et al.*, 1981] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7(1):17–41, 1981.

[Moré, 1978] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Proceedings Dundee 1977*. Springer Verlag, Berlin, 1978. Lecture Notes in Mathematics.

[Moré, 1983] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 258–287. Springer Verlag, Berlin, 1983.

[Munksgaard and Reid, 1983] N. A. Munksgaard and J. K. Reid. NS02, a Fortran subroutine for solving sparse sets of non-linear equations by Powell's dog-leg algorithm. Technical Report R11047, AERE Harwell Laboratory, Harwell, UK, 1983.

[Murray and Wright, 1976] W. Murray and M. H. Wright. Efficient line search algorithms for the logarithmic barrier function. Technical Report SOL76-18, Department of Operations Research, Stanford University, Stanford, California 94305, USA, 1976.

[Murray and Wright, 1992] W. Murray and M. H. Wright. Line search procedures for the logarithmic barrier function. Numerical analysis manuscript 92-01, AT&T Bell Laboratories, Murray Hill, USA, 1992.

[Murtagh and Saunders, 1987] B. A. Murtagh and M. A. Saunders. MINOS 5.1 USER'S GUIDE. Technical Report SOL83-20R, Department of Operations Research, Stanford University, Stanford, USA, 1987.

[Nash and Sofer, 1991] S. G. Nash and A. Sofer. A general-purpose parallel algorithm for unconstrained optimization. *SIAM Journal on Optimization*, 1:530–547, 1991.

[Nash and Sofer, 1993] S. G. Nash and A. Sofer. A barrier method for large-scale constrained optimization. *ORSA Journal on Computing*, 5(1):40–53, 1993.

[Nash *et al.*, 1993] S. G. Nash, R. Polyak, and A. Sofer. A numerical comparison of barrier and modified barrier methods for large-scale constrained optimization. Technical Report 93-02, Department of Operations Research, George Mason University, Fairfax, Virginia 22030, 1993.

[Nemhauser *et al.*, 1989] G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd. *Optimization*, Volume 1 of *Handbooks in Operations Research and Management Science*. North-Holland, Amsterdam, 1989.

[Omojokun, 1991] E. O. Omojokun. *Trust region algortihms for optimization with nonlinear equality and inequality constraints*. PhD thesis, Department of Computer Sciences, University of Colorado, Boulder, USA, 1991.

[Polyak, 1992] R. Polyak. Modified barrier functions (theory and methods). *Mathematical Programming*, 54(2):177–222, 1992.

[Powell, 1970] M. J. D. Powell. A new algorithm for unconstrained optimization. In J. B. Rose, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*. Academic Press, New York, 1970.

[Powell, 1975] M. J. D. Powell. Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming, 2*. Academic Press, New York, 1975.

[Powell, 1982] M.J.D. Powell. Extensions to subroutine VF02. In R.F. Drenick and F. Kozin, editors, *Systems Modelling and Optimization. Lecture notes in control and Information sciences 38*, pages 529 – 538. Springer-Verlag, Berlin, 1982.

[Powell, 1992] M. J. D. Powell. Some convergence properties of the shifted log barrier method for linear programming. Technical Report DAMTP NA7, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, UK, 1992.

[Robinson, 1972] S. M. Robinson. A quadratically convergent algorithm for general nonlinear programming problems. *Mathematical Programming*, 3:145–156, 1972.

[Rosen and Kreuser, 1972] J. B. Rosen and J. Kreuser. A gradient projection algorithm for nonlinear constraints. In F. A. Lootsma, editor, *Numerical Methods for Nonlinear Optimization*, pages 39–43. Academic Press, London, 1972.

[Saludjian, 1993] L. Saludjian. Etude d'une version parallèle de Lancelot dans l'environment de programmation par tranferts de message PVM. Rapport de stage de diplome d'études approfondies, Toulouse, France, 1993.

[Sartenaer, 1993] A. Sartenaer. A class of trust region methods for nonlinear network optimization problems, including numerical experiments. Technical Report 93/21, Department of Mathematics, FUNDP, Namur, Belgium, 1993. To appear in SIAM Journal on Optimization.

[Schittkowski, 1987] K. Schittkowski. *More Test Examples for Nonlinear Programming Codes*, Volume 282 of *Lecture notes in economics and mathematical systems*. Springer Verlag, Berlin, 1987.

[Schlick, 1993] T. Schlick. Modified Cholesky factorizations for sparse preconditioners. *SIAM Journal on Scientific and Statistical Computing*, 14(2):424–445, 1993.

[Schnabel and Chow, 1991] R.B. Schnabel and T.-T. Chow. Tensor methods for unconstrained optimization using second derivatives. *SIAM Journal on Optimization*, 1(3):293–315, 1991.

[Schnabel and Eskow, 1991] R. B. Schnabel and E. Eskow. A new modified Cholesky factorization. *SIAM Journal on Scientific and Statistical Computing*, 11:1136–1158, 1991.

[Schnabel and Frank, 1984] R. B. Schnabel and P. D. Frank. Tensor methods for nonlinear equations. *SIAM Journal on Numerical Analysis*, 21(5):815–843, 1984.

[Schnabel, 1994] R. B. Schnabel. Parallel nonlinear optimization. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Schrady and Choe, 1971] D. A. Schrady and U. C. Choe. Models for multi-item continuous review inventory policies subject to constraints. *Naval Research Logistics Quarterly*, 18:451–463, 1971.

[Shanno and Phua, 1980] D. F. Shanno and K. H. Phua. Remark on algorithm 500. *ACM Transactions on Mathematical Software*, 6:618–622, 1980.

[Shanno, 1994] D. F. Shanno. Algorithms for linear programming. In E. Spedicato, editor, *Algorithms for continuous optimization: the state of the art.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[Sklar and Armstrong, 1993] M. G. Sklar and R. D. Armstrong. Lagrangian approach for large-scale distance value estimation. *Computers Operations Research*, 20:83–93, 1993.

[Smith and Lasdon, 1992] S. Smith and L. Lasdon. Solving large sparse nonlinear programs using GRG. *ORSA Journal on Computing*, 4:2–15, 1992.

[Soares *et al.*, 1993] J. Soares, J. J. Júdice, and F. Facchinei. An active set Newton's algorithm for large-scale nonlinear programs with box constraints. Technical report, Universidade de Coimbra, 300 Coimbra, Portugal, 1993.

[Sorensen, 1981] D. C. Sorensen. An example concerning quasi-Newton estimates of a sparse Hessian. *SIGNUM Newsletter*, 16:8–10, 1981.

[Steihaug, 1983a] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[Steihaug, 1983b] T. Steihaug. Local and superlinear convergence for truncated iterated projection methods. *Mathematical Programming*, 27:199–223, 1983.

[Toint and Tuyttens, 1990] Ph. L. Toint and D. Tuyttens. On large scale nonlinear network optimization. *Mathematical Programming, Series B*, 48(1):125–159, 1990.

[Toint and Tuyttens, 1992] Ph. L. Toint and D. Tuyttens. LSNNO: a Fortran subroutine for solving large scale nonlinear network optimization problems. *ACM Transactions on Mathematical Software*, 18(3):308–328, 1992.

[Toint, 1981a] Ph. L. Toint. A sparse quasi-Newton update derived variationally with a non-diagonally weighted Frobenius norm. *Mathematics of Computation*, 37(156):425–433, 1981.

[Toint, 1981b] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57- -88. Academic Press, London, 1981.

[Toint, 1983] Ph. L. Toint. Test problems for partially separable optimization and results for the routine PSPMIN. Technical Report 83/4, Department of Mathematics, FUNDP, Namur, Belgium, 1983.

[Toint, 1987a] Ph. L. Toint. On large scale nonlinear least squares calculations. *SIAM Journal on Scientific and Statistical Computing*, 8(3):416–435, 1987.

[Toint, 1987b] Ph. L. Toint. VE10AD, a routine for large scale nonlinear least squares. *Harwell Subroutine Library*, 1987.

[Vanderbei and Carpenter, 1993] R. J. Vanderbei and T. J. Carpenter. Symmetric indefinite systems for interior point methods. *Mathematical Programming, Series A*, 58(1):1–32, 1993.

[Werbos, 1988] P. Werbos. Backpropagation: past and future. In *Proceedings of the 2nd International Conference on Neural Networks*. IEEE, New York, 1988.

[Wright, 1991] M. H. Wright. Optimization and large scale computation. In J.P. Mesirov, editor, *Very Large Scale Computation in the* 21^{st} *Century*, pages 341–407. SIAM, Philadelphia, 1991.

[Wu, 1993] Z. Wu. The effective energy transformation scheme as a general continuation approach to global optimization with application to molecular conformation. Technical Report CTC93TR143, Advanced Computing Research Institute, Cornell Theory Center, Cornell University, Ithaca, USA, 1993.

[Zenios and Mulvey, 1986] S. A. Zenios and J. M. Mulvey. Nonlinear network programming on vector supercomputers: A study on the CRAY X-MP. *Operations Research*, 34(5):667–682, 1986.

[Zenios and Mulvey, 1988] S. A. Zenios and J. M. Mulvey. Vectorization and multitasking of nonlinear network programming algorithms. *Mathematical Programming, Series A*, 42(2):449–470, 1988.

[Zenios and Pinar, 1989] S. A. Zenios and M. C. Pinar. Parallel block-partitioning of truncated Newton for nonlinear network optimization. Technical Report 89-09-08, Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, USA, 1989.

[Zenios, 1989] S. A. Zenios. Parallel optimization: current status and an annotated bibliography. *ORSA Journal on Computing*, 1:20–43, 1989.

[Zou *et al.*, 1993] X. Zou, I.M. Navon, M. Berger, K. H. Phua, T. Schlick, and F. X. LeDimet. Numerical experience with limited-memory quasi-Newton and truncated Newton methods. *SIAM Journal on Optimization*, 3(3):582–608, 1993.