AMO - Advanced Modeling and Optimization, Volume 10, Number 1, 2008

A Simple Decomposition based SQP Algorithm for Large Scale Nonlinear Programming

Mehdi Lachiheb

Faculté des Sciences de Gabès, Cité Erriadh 6072, Gabès, Tunisie.

lachihebm@yahoo.ca

Hichem Smaoui

Laboratoire de Systèmes et Mécanique Appliquée, Ecole Polytechnique de Tunisie.

hismaoui@yahoo.fr

Abstract: The simple decomposition, originally developed for quadratic programming, is incorporated into an SQP algorithm in order to handle large scale nonlinear programming problems. The resulting algorithm is tested on truss optimum design problems and some analytical problems. Results indicate excellent accuracy and considerable computational advantage in favor of the proposed algorithm with respect to existing reference codes, especially in problems where the number of active constraints approaches the number of variables. Moreover, analysis of the evolution of the optimum set of extreme points of the sequence of quadratic programming problems led to the development of a procedure for initiating the decomposition with a whole set of extreme points that further enhances the computational performance of the proposed algorithm.

Keywords: sequential quadratic programming, nonlinear programming, simple decomposition, large scale, extreme point, structural optimization.

[0]AMO - Advanced Modeling and Optimization. ISSN: 1841-4311

1 Introduction

The sequential quadratic programming method (SQP) [Powell, 1978, Shanno and Phua, 1989, Lawrence and Tits, 2001, Schittkowski, 2005], developed in the late seventies, is recognized as one of the best methods available today for solving general nonlinear programming problems of the form

$$P \qquad \begin{cases} \quad \min \ f(x) \\ g_i(x) \leq 0 \quad i \in I \\ h_i(x) = 0 \quad i \in L \\ \quad x \in IR^n \end{cases} \tag{1}$$

where $I = \{1, 2, ..., m\}$ and $L = \{1, 2, ..., l\}$. It consists of iteratively solving a sequence of quadratic programming problems. Consequently, in any implementation of the SQP algorithm for large scale nonlinear programming problems [Zillober,2004], one should face the issue of solving large quadratic programming problems (QP)s. In the code $NLPQL$ [Powell, 1983, Schittkowski, 1985,2004], Schittkowski used a dual algorithm to solve the QP, whereas in [Boggs, Kearsley, Tolle, 1999] Boggs, A.J. Kearsley and J.W. Tolle adopted an interior penalty method. In the present work, Sacher's decomposition technique [Sacher, 1980], derived from Hohenbalken's simple decomposition [Hohenbalken, 1977], is proposed for solving the QP. It consists in transforming the original quadratic programming problem, whose variables form the space vector, into a problem whose variables are the coefficients of the convex combinations expressing the space vector in terms of the extreme points of the feasible set. Solving a quadratic programming problem is then achieved iteratively via the repeated solution of two problems: a master problem and a subproblem. In his original implementation [Sacher, 1980], Sacher used Lemke's method to solve the master problem. Later, Ben Daya [Ben Daya, 1994] applied a logarithmic barrier function to improve convergence when the feasible domain of the master problem is bounded.
The objective of the present work is to take advantage of Sacher's decomposition by incorporating it into the SQP algorithm in order to enhance its large scale capabilities [Mulvey, Zenios and Ahlfeld, 1990, Mar'in, 1995].
In the present paper, first, the barrier function used in solving the master problem [Ben Daya, 1994] is modified to accommodate unbounded feasible domain by expressing the solutions as combinations of extreme points

and extreme rays. The decomposition algorithm has been subjected to a significant number of tests on examples of QPs. A number of these test problems have been constructed in a way to exhibit specific features such as ill-conditioning of the objective function [Lachiheb, 1997]. Second, the decomposition method thus implemented is integrated into a sequential quadratic programming algorithm to form a general nonlinear programming code [Lachiheb, 1997] that will be denoted $SQPD$. The latter has been validated through a number of numerical tests, each problem being subjected to many runs using different starting solutions. Then the evolution of the optimum set of extreme points (SEP) from a SQP iteration to another is analyzed. Examination of this evolution led to the development of a procedure that aims at reducing the computational effort devoted to the generation of intermediate extreme points. The underlying idea consists in initiating the decomposition process with a whole SEP instead of a single extreme point. The initial SEP is determined from the results of the preceding iteration of the SQP sequence without solving a series of master problems and subproblems. Finally, numerical results are presented for several nonlinear programming example problems including truss optimum design problems and some analytical problems. These results illustrate the performance of the original and the modified $SQPD$ algorithms and comparison is made with the reference code $NLPQL$.

2 Sequential quadratic programming

The sequential quadratic programming method combines the advantages of variable metric methods for unconstrained optimization with the rapid convergence of Newton's method for solving nonlinear systems of equations [Fiacco and McCormik, 1968]. Newton's method is known to be quadratically convergent but requires second order derivatives. Like variable metric algorithms for unconstrained minimization, the SQP method uses only first order derivatives and exhibits superlinear convergence [Powell, 1978]. It is based on the works of [Biggs, 1975], [Han, 1977] and [Powell, 1977]. The algorithm consists in solving a sequence of quadratic programming problems of the form

$$QP_k \left\{ \begin{array}{c} \min\ Q(d) = \frac{1}{2} d^t B_k d + d^t \nabla f(x^k) \\ \nabla\, g_i(x^k)^t d +\ g_i(x^k) \leq 0 \quad i \in I \\ \nabla h_i(x^k)^t d + h_i(x^k) = 0 \quad i \in L \\ d \in I\!R^n \end{array} \right. \tag{2}$$

where B_k is an approximation of the Hessian $L(x^*, \lambda^*, \mu^*)$ of the Lagrangian function :

$$l(x, \lambda^*, \mu^*) = f(x) + \sum_{i \in I} \lambda_i^* g_i(x) + \sum_{j \in L} \mu_j^* h_j(x) \tag{3}$$

over the set of feasible directions at the solution x^* of problem P, λ^* and μ^* being the optimal Lagrange multipliers.

3 Sacher's simple decomposition:

Sacher's simple decomposition [Sacher, 1980, Shetty 1988] is a method for solving quadratic programming problems of the form:

$$\left\{ \begin{array}{c} \min\ \frac{1}{2} x^t B x\ +\ c^t x \\ A_1 x \geq b_1 \\ A_2 x = b_2 \\ x \geq 0 \end{array} \right. \tag{4}$$

where $x = (x_1, x_2, ..., x_n) \in I\!R^n$ is the vector of variables, B is a $n \times n$ positive semi-definite matrix, A_1 and A_2 are respectively $m_1 \times n$ and $m_2 \times n$ matrices, c, b_1 and b_2 are vectors of dimensions n, m_1 and m_2 respectively.
Let $S = \{x \in I\!R^n,\ A_1 x\ \geq\ b_1,\ A_2 x\ =\ b_2 \text{ and } x\ \geq\ 0\}$ be the feasible set for problem (4). S is a convex polytope [Rockafellar, 1970], therefore there exist p extreme points $x^1, x^2, x^3, ..., x^p$ $(p \geq 1)$ and q extreme rays $d^1, d^2, d^3, ..., d^q$, $(q \geq 0)$ such that

$$\forall\ x\ \in\ S,\ \exists\ u_1, ..., u_p,\ \ v_1, ..., v_q\ \in I\!R_+\ /$$
$$\sum_{i=1}^{p} u_i = 1 \quad \text{and} \quad x\ =\ \sum_{i=1}^{p} u_i x^i\ +\ \sum_{i=1}^{q} v_j d^j\ .$$

or, in matrix notation

$$x = \quad Uu \quad + \ Vv \tag{5}$$

where

$$U = \begin{pmatrix} x_1^1 & x_1^2 & ... & x_1^p \\ \\ x_2^1 & x_2^2 & ... & x_2^p \\ ... & ... & ... & ... \\ x_n^1 & x_n^2 & ... & x_n^p \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} d_1^1 & d_1^2 & ... & d_1^q \\ \\ d_2^1 & d_2^2 & ... & d_2^q \\ ... & ... & ... & ... \\ d_n^1 & d_n^2 & ... & d_n^q \end{pmatrix}.$$

For simplicity of notation, we introduce the $n \times (p+q)$ matrix $W = (U\, , V)$ and the $(p+q)$-vector $w = \begin{pmatrix} u \\ v \end{pmatrix}$ so that (5) can be written in the form $x = Ww$. Substituting Ww for x in problem (4) gives rise to an equivalent problem, called master problem (MP), defined by

$$MP \quad \begin{cases} \min \ \frac{1}{2} w^t Q w \ + \ s^t w \\ \sum\limits_{i=1}^{p} w_i \ = 1 \\ \qquad w \geq 0 \end{cases} \tag{6}$$

where $Q = W^t B W$ is a $(p+q) \times (p+q)$ positive semi-definite matrix and $s = W^t c$ is a $(p+q)$-vector. Clearly, the master problem involves a single constraint, moreover, if the number of extreme points needed to express the solution is low, the problem size may become smaller than that of the original QP (4). The basic idea of the algorithm is to operate iteratively in subspaces generated by a small number of extreme points which are updated via the solution of a linear programming problem, called subproblem (SP),

$$SP \quad \begin{cases} \min \ h^t x \\ A_1 x \ \geq b_1 \\ A_2 x \ = b_2 \\ \quad x \ \geq 0 \end{cases} \tag{7}$$

where h is the gradient of the objective function of problem (4) at the solution of the current master problem. The subproblem also provides the termination test for the algorithm.

3.1 Simple decomposition algorithm :

Sacher's simple decomposition algorithm can be summarized in the following steps.

- **Step 1** : Let U and V be two matrices made up columnwise of extreme points and extreme rays respectively. U has at least one column whereas V may be empty.

• **Step 2** : Solve the master problem *MP*. If it is unbounded the problem (4) is also unbounded. Otherwise, let $\begin{pmatrix} u \\ v \end{pmatrix}$ denote the solution of the master problem and let

$$\tilde{x} = Uu + Vv.$$

• **Step 3** : Solve the subproblem (7) where $h = BUu + BVv + c = B\tilde{x} + c$. If the solution of *SP* is bounded, then it must coincide with an extreme point which will be denoted by x^k. Otherwise let d^k be a feasible descent direction ($h^t d^k < 0$).

• **Step 4** : If *SP* is bounded and has a solution x^k such that $h^t\tilde{x} = h^t x^k$, then $\tilde{x}$ is the solution of problem (4). Otherwise go to Step 5.

• **Step 5** : If there exists $i \in IN \,/\, u_i = 0$ (resp. $v_i = 0$) then eliminate extreme point x^i (resp. extreme ray d^i). If subproblem (7) is bounded, then replace U *by* (U, x^k). Otherwise replace V by (V, d^k). Go to Step 1.

3.2 Convergence of the simple decomposition algorithm

When the feasible set of the problem (4) is bounded, that of the master problem is always the convex hull of a set of affinely independent extreme points, therefore, the dimension of the master problem never exceeds $n + 1 - m'$, where m' is the rank of the Jacobian of the active constraints, including equalities. This, combined with the descent property that results from the subproblem formulation, implies finite convergence of the simple decomposition [Lachiheb, 1997]. A particularly interesting case is that of problems having a number of active constraints close to the number of variables. With these problems the dimension of the master problem is small, which makes the decomposition method highly efficient.

3.3 Solution of the master problem:

The structure of the master problem makes it suitable for solution by a penalty method. When the feasible set for the original quadratic problem is bounded the vector w in problem (6) is made up solely of the components u_i verifying $\sum_{i=1}^{p} u_i = 1$. The barrier function used in [Ben Daya, 1994] is:

$$K(x,r) = -r \sum_{i=1}^{n_k} \log x_i .$$

where n_k is the current number of extreme points and extreme rays and r, a positive real number, is the penalty coefficient. The above function is not usable in general if the feasible set is unbounded . However, it is applicable under the assumption of positive definiteness of matrix B.

Remarks:

- in case the function f is not strictly convex one can choose another penalty function $K(x,r)$ defined by

$$K(x,r) = -r \sum_{i=1}^{n_k} H(x_i)$$

where

$$H(x_i) = \begin{cases} \log x_i & \text{if } x_i \leq 1 \\ 1 - \frac{1}{x_i} & \text{if } x_i \geq 1 \end{cases}$$

which is continuous and differentiable over S.

- an advantage of the adopted choice for the penalty function is in that the barrier function is strictly convex even when the original function is nonconvex. This ensures uniqueness of the optimum for any value of r. In the following, the objective function of the problem (4) is assumed to be strictly convex. The penalized problem is written as

$$MPP_r \quad \begin{cases} \min \ \frac{1}{2} w^t Q w + s^t w - \ r \sum_{i=1}^{n_k} \log w_i \\ \sum_{i=1}^{p} w_i \ = 1 \end{cases} . \tag{8}$$

For every solution $w = \begin{pmatrix} u \\ v \end{pmatrix}$ let

- r, a positive real number, is the penalty coefficient.
- D denote the diagonal matrix of dimension n_k having w_i as components,
- e denote the n_k vector whose first p components are ones and the remaining are zeroes,
- $f_r(w) = \frac{1}{2} w^t Q w \ + \ s^t w \ - \ r \sum_{i=1}^{n_k} \log w_i$,
- $g_r(w) \ = \ \nabla f_r(w) \ = \ Qw \ + s \ - r \ D^{-1} e$,
- $H_r(w) \ = \nabla^2 f_r(w) \ = Q \ + \ r \, D^{-2}$.

Lemma :[Ben Daya, 1994]

For each penalty coefficient $r_j > 0$, let λ_j be the Lagrange multiplier associated with the unique constraint of problem MPP_{r_j}. Then

$$\lambda_j = \frac{e^t H_{r_j}^{-1} g_{r_j}}{e^t H_{r_j}^{-1} e}$$

and the Newton direction for problem MP_{r_j} at w is given by:

$$d_j = -H_{r_j}^{-1}(g_{r_j} - \lambda_j e).$$

4 Perturbation of extreme points

4.1 Extreme point characterization

In case the feasible set

$$S = \{x \in IR^n / \;\; A_1 x \geq b_1, \, A_2 x = b_2 \text{ and } \; x \geq 0\}. \qquad (9)$$

of problem (4) is unbounded one may change it into a bounded set without altering the optimum solution, simply by imposing supplementary constraints $x_i \leq a, i = 1, ..., n$ where a is a sufficiently large real number. In the following, the assumption of bounded feasible set will be made. Feasible solutions are, therefore, written as convex combinations of extreme points only. The feasible set of a generic quadratic programming problem in the SQP sequence is defined by (9). In order to characterize the extreme points of S we introduce slack variables and rewrite it as $S = \{x \in IR^n / \; \exists \; h \in IR^{m_1} \; / \; (x, h) \in H\}$ where

$$H = \left\{ (x, h) \in IR^n \times IR^{m_1} / \; A \begin{pmatrix} x \\ h \end{pmatrix} = b \, , \quad x \geq 0 \text{ and } \; h \geq 0 \right\} ,$$

$A = \begin{pmatrix} A_1 & -I \\ A_2 & 0 \end{pmatrix}$, $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, I denoting the $m_1 \times m_1$ identity matrix. Thus, each extreme point is defined by an $m \times m$, $(m = m_1 + m_2)$ nonsingular submatrix of A, or simply by a set of m columns of A.

4.2 Influence of conditioning

Let B be a nonsingular submatrix of A and x the solution of the equation $Bx = b$. A small perturbation in the matrix A and the right hand side b results in a perturbation in the set H, and possibly in a change in the topology and number of its extreme points. The following cases may occur for a given extreme point characterized by a matrix B:

i/ the perturbed matrix $B + \delta B$ is singular, therefore no extreme point can be associated to it. In other words, at least one extreme point leaves the SEP as a result of the perturbation. This may happen when the matrix B is ill-conditioned
ii/ The matrix $(B + \delta B)$ is nonsingular and the equation

$$(B + \delta B)x = b + \delta b$$

has no nonnegative solution, which implies that the extreme point associated with matrix B transforms into a point which is not a vertex of the perturbed domain $H' = \{x \in IR^n / \ (A + \delta A)\, x \geq b + \delta b \text{ and } x \geq 0\}$. In this case at least one extreme point enters the SEP. This may occur either at a nondegenerate extreme point with an ill-conditioned associated matrix B, or at a degenerate point independently of the conditioning of its associated matrix.
iii/ The matrix $(B + \delta B)$ is nonsingular and the equation

$$(B + \delta B)x = b + \delta b$$

has a nonnegative solution, which defines an extreme point $(x^i + \delta x^i)$. If the matrix B is well conditioned the perturbed extreme point should be close to x^i according to the following proposition.
Proposition 1 [Ciarlet, 1982]. Let $\| \ . \|$ denote a subordinate matrix norm. If $\|\delta B\| < \frac{1}{\|B\|}$ then

$$\tfrac{\|\delta x\|}{\|x\|} \leq \tfrac{1}{1 - \|B^{-1}\| \|\delta B\|} (Cond(B) (\tfrac{\|\delta B\|}{\|B\|} + \tfrac{\|\delta b\|}{\|b\|}))$$

where

$$Cond(B) = \|B^{-1}\| \, \|B\| \, .$$

4.3 Approximation of the optimum *SEP*

The solution of the quadratic programming problem by the standard simple decomposition algorithm has been subjected to testing on many example problems. Examination of the variations of the extreme points through the SQP iterations has shown that, in some problems, particularly those exhibiting ill-conditioning, the number of extreme points getting in and out of the SEP is very large. Considering that the generation of each extreme point requires the solution of a large LP problem in addition to that of the master problem, the overall computational effort could be improved significantly if the number of extreme point generations were reduced. On the other hand, it has been noted that, in most cases and especially at the tail of the sequence, to each point in the optimum SEP of problem QP_k is associated a point in the optimum SEP of problem QP_{k+1} defined by the same columns in the coefficient matrix. In such cases the $k+1^{st}$ optimum SEP can be viewed as the result of the k^{th} optimum SEP by a smooth mapping T. This leads to the idea of obtaining the entire optimum SEP for a new QP directly from the previous one, at least in an approximate way, in general. In an attempt to construct an approximation of the $k+1^{st}$ optimum SEP, the following approach is considered. Let $\{x^i, i = 1, ..., n_k\}$ be the optimum SEP for the k^{th} iteration. For each point x^j, one seeks a corresponding extreme point, for the feasible set S_{k+1}, that is characterized as the closest one to x^j. The new extreme point, denoted by y^j, is sought as the solution of the problem:

$$LSP_{jk} \quad \begin{cases} \min \sum\limits_{i \in L_j} x_i \\ x \in S^{k+1} \end{cases} \tag{10}$$

$L_j = \{i \in IN \ / \ \left\| x_i^j \right\| \leq \epsilon\}$, where ϵ is a small nonnegative real number.
A drawback of the above formulation is that, due to ill-conditioning or degeneracy, the new points $y^i, i = 1, ..., n_k$ are not necessarily affinely independent, which may cause the number of points in the SEP to exceed the limit $n + 1 - m'$ in subsequent steps of the decomposition procedure. The largest affinely independent subset can be determined by applying the simplex algorithm to the following problem:

$$R_k \quad \begin{cases} \min \ x \ = \sum\limits_{i=1}^{n_k} u_i. \\ \sum\limits_{i=1}^{n_k} u_i y^i = x^* \\ \sum\limits_{i=1}^{n_k} u_i = 1 \\ u \geq 0 \end{cases} \tag{11}$$

where $x^* = \sum\limits_{i=1}^{n_k} u_i^* y^i$, u_i^* are the components of the optimum solution of the master problem corresponding to the SEP $\{y^i, \ i = 1, ...n_k\}$. The set of independent extreme points is obtained by retaining solely extreme points whose corresponding optimal coefficients are positive. The resulting set forms the initial group of extreme points for problem QP_{k+1}.

5 Numerical Examples

The proposed algorithm $SQPD$ has been applied in its original and modified versions to a number or test problems. A representative selection of these problems is presented in this section. Comparison of computing times is made at the end of the section.

5.1 Problem Description and Results

5.1.1 Powell's Problem

Powell's problem [Powell, 1978] is an example with a small number of variables and exhibiting pronounced nonlinearity. Table 1 presents the sequence of optimum SEP corresponding to a run of the SQP algorithm started at the solution $x_0 = (0, -2, 2, 0, -1)$ using the unmodified version of the simple decomposition. It can be seen that the maximum number of extreme points used at a given step is 4, that is less than $n + 1 = 6$. The basic columns stabilize from the third iteration for extreme points x^3, from the fourth iteration for x^1 and from the sixth for x^4. It can be noted that the latter leaves the optimum SEP at iteration 4 and reenters it at the sixth iteration. The optimum solution obtained is $x^* = (-0.699034,\ -0.869963,\ 2.789922,\ 0.6968791,\ -0.69657065)$ and the objective value is 0.4388502. On the other hand it should be noted that convergence of the SQP sequence is achieved within 8 iterations with a tolerance of 10^{-5} on the norm of direction d, *i.e.* the same number of iterations as reported in [Powell, 1978]. Similarly, Table 2 presents the sequence of optimum SEP using the starting point $x_0 = (-2, 2, 2, -1, -1)$. In this example, the basic columns are seen to stabilize from the first iteration for all extreme points. Extreme point x^2 leaves the SEP at the fourth iteration. The optimum solution obtained is $x^* = (-1.71714,\ 1.59571,\ 1.82723,\ -0.76364,\ -0.76364)$ and the objective value is 0.0539495.

5.1.2 Truss Design Example problems

In this section, examples of optimum design problems of truss structures with a wide range of dimensions are considered. The typical design problem consists in minimizing the self weight of the truss subject to limitations on the stress in every member both in compression and in tension and to lower bound restraints on the cross sectional areas which constitute the design variables.

Iter.	1	2	3	4	5	6	7	8
	.00000	0.0000	.00000	.00000	.00000	.00000	.00000	0.0000
	49.416	49.570	59.557	68.526	75.862	82.122	82.283	82.282
$x1$	24.166	27.685	35.386	42.095	49.420	53.821	53.782	53.770
	39.100	25.445	12.031	.00000	.00000	.00000	.00000	.00000
	0.0000	0.0000	0.0000	.74450	15.535	25.021	24.958	24.933
	$1.E+5$	$1.E+5$	309.49	184.783	146.59	.00000	.00000	.00000
	49.416	49.570	.00000	.00000	.00000	82.122	82.283	82.282
$x2$	50024.	44132.	23028.	13505.	5334.4	.00000	.00000	27.242
	20039.	49049.	64808.	76366.	90345.	16968.	90231.	$1.E+5$
	$1.E+5$	$1.E+5$	$1.E+5$	$1.E+5$	$1.E+5$	16892.	90155.	99975.
	.00000	0.0000	309.49	184.78	146.59	127.80	127.43	127.44
	49.416	49.570	.00000	.00000	.00000	.00000	.00000	.00000
$x3$	50024.	44132.	55.283	46.717	44.959	44.336	44.194	44.183
	20039.	49049.	47.509	45.336	41.332	38.406	38.554	38.573
	$1.E+5$	$1.E+5$	.00000	.00000	.00000	0.0000	.00000	.00000
	$1.E+5$	$1.E+5$				50.599	50.112	50.049
	49.416	49.570				49.609	49.927	49.967
$x4$	24.166	27.699				50.115	50.021	50.009
	39.100	25.449				.00000	.00000	.00000
	0.0000	0.0000				.00000	.00000	.00000
$\|d\|$	0.8074	0.3264	0.2187	0.1700	0.0679	0.0044	0.0002	.00004

Table 1. Sequence of optimum SEP and $\|d\|$ for Powell's problem $x_0 = (0, -2, 2, 0, -1)$.

$Iter.$	1	2	3	4	5
	22.3636	24.2513	24.3366	23.1894	$23,0914$
	77.7196	78.7143	79.3742	81.0056	$81,0056$
$x1$	.000000	.000000	.000000	.000000	.000000
	108.712	108.738	108.171	106.168	105.985
	.000000	.000000	.000000	.000000	.000000
	100.083	94.8945	93.6834		
	.000000	000000	.000000		
$x2$	142.486	137.057	135.109		
	.000000	.000000	.000000		
	82.8055	84.7330	86.0907		
	22.3636	24.2513	24.3366	23.1894	23.0914
	77.7196	78.7143	79.3742	81.0056	81.1590
$x3$	.000000	000000	.000000	.000000	.000000
	.000000	000000	.000000	.000000	.000000
	108.712	108.738	108.171	106.168	105.985
	100.083	94.8945	93.6834	93.2352	93.1792
	.000000	000000	.000000	.000000	.000000
$x4$	142.486	137.057	135.109	130.634	130.233
	82.8054	84.7330	86.0907	90.0489	90.3946
	.000000	.000000	.000000	.000000	.000000
$\|d\|$	0.30900	0.02544	0.02190	0.00625	0.00006

Table 2. Sequence of optimum SEP and $\|d\|$ for Powell's problem $x_0 = (-2, 2, 2, -1, -1)$.

Ten Bar Truss. In this example the optimum design problem for the ten bar truss structure depicted in Figure 1 is considered. The detailed problem statement is given in [Kirsch, 1981]. The allowable stresses and the lower bound on the cross sectional area are respectively $\pm 2500 kips$ and 0.1 in^2. It can be noted that the problem statement is independent of the Young modulus which does not need to be specified. The problem is solved by the SQP algorithm using the unmodified simple decomposition. The sequence converges within 6 iterations with a tolerance of 10^{-6} on $\|d\|$. The optimal solution obtained is $x^* = (7.937867,\ 0.1,\ 8.0621,\ 3.9379,\ 0.1,\ 0.1,\ 5.7447,\ 5.5690,\ 5.5690,\ 0.1) in^2$ and the optimum volume is $15931,8 in^3$. Table 3 shows the sequence of optimum SEP. It is interesting to note that, except for the first iteration, the optimum SEP reduces to a singleton. Indeed, the number m' of active constraints, including lower bound constraints on the variables, is 10, so that $n + 1 - m' = 1$. As a consequence, there is no master problem to solve. Moreover, the unique extreme point corresponds to a constant set of basic columns with respect to both the original design variables and the slack variables.

Iter.	1		2	3	4	5	6
	$x1$	$x2$	$x1$	$x1$	$x1$	$x1$	$x1$
	4.76881	4.35332	6.39218	7.57217	7.82922	7.83786	7.83787
	.000000	.000000	.000000	.000000	.000000	.000000	.000000
	8.09764	8.89494	7.88101	7.96076	7.96213	7.96213	7.96213
	3.52469	3.39200	3.79158	3.83744	3.83787	3.83787	3.83787
	.000000	.000000	.000000	.000000	.000000	.000000	.000000
	.000000	.000000	.000000	.000000	.000000	.000000	.000000
	4.39436	5.29781	5.60752	5.64397	5.64472	5.64472	5.74472
	4.18233	3.25080	4.66185	5.36208	5.46714	5.46899	5.46899
	5.19629	4.99725	5.43361	5.46885	5.46899	5.46899	5.46899
	.000000	.000000	.000000	.000000	.000000	.000000	.000000
$\|d\|$	2.090		2.008	1.210	0.0257	0.0086	0.00001

Table 3. Sequence of optimum SEP and $\|d\|$ for ten bar truss problem.

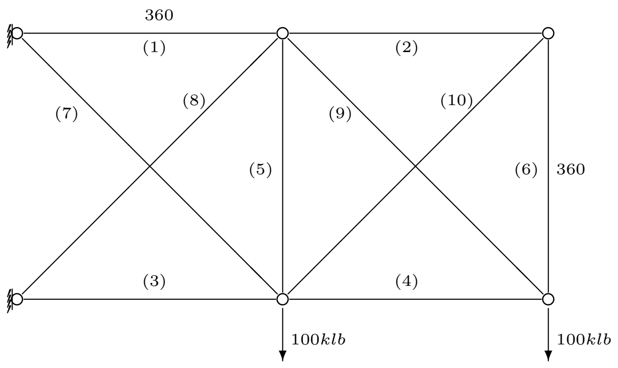

Figure 1. Ten bar truss (unit: inch)

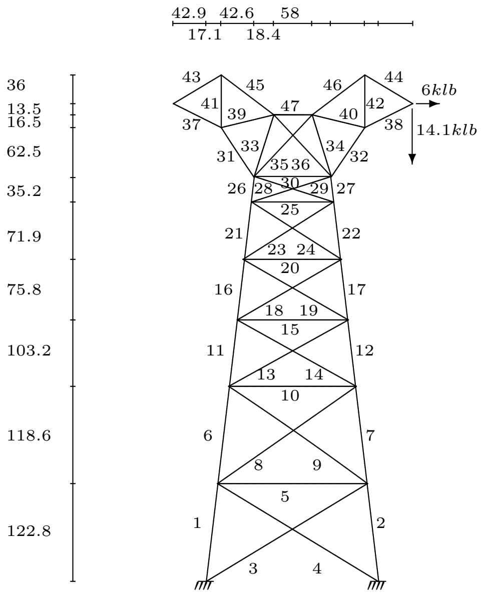

Figure 2. Forty-seven-bar transmission tower (unit: inch)

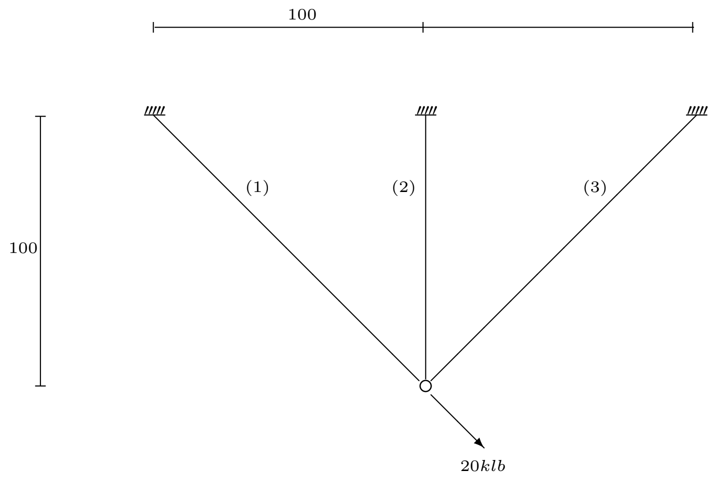

Figure 3. Three bar truss (unit: inch)

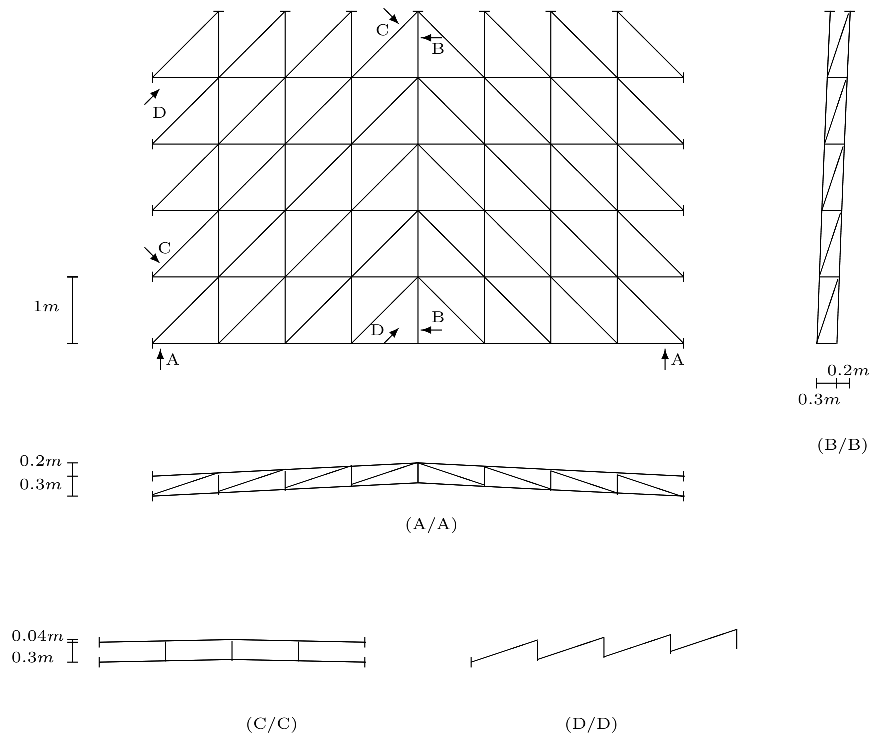

Figure 4. Truss framing Structure

Forty-seven-bar transmission tower. In this example a larger truss structure is considered and design variable linking is applied so that the number of design variables is reduced to 27. The detailed problem statement is given in [Kirsch, 1981] and the structure is shown in Figure 2.
The truss is to be designed for minimum self weight subject to stress constraints given an allowable stress limit of 35 kips, and 0.1 in^2 minimum gage restraint on the cross sectional areas. Starting from an initial design with all cross sections at 10 in^2 and using the $SQPD$ algorithm, the solution is obtained within 10 SQP iterations. The optimal solution is $x^* = (1.340856,$ $0.10, 0.10, 1.274094, 0.10, 0.10, 1.180685, 0.111531, 0.10, 1.066225, 0.121519,$ $0.10, 0.931220, 0.140893, 0.10, 0.768537, 0.232367, 0.5335577, 0.853049, 0.337046,$ $0.342363, 0.140052, 0.216507, 0.626915, 0.517970, 0.571995, 0.428604) in^2$ and the optimum volume is 2172.73 in^3. As observed in the previous example, the optimum SEP is a singleton except for the first iteration. This is in agreement with the statement that the number of retained extreme points is governed by the number $n + 1 - m'$. Indeed, in both problems the number of active constraints, including side constraints, is equal to the number of independent design variables, that is $n + 1 - m' = 1$.
Three Bar Truss. The structure and loading of this example are described in Figure 3. The design problem consists of finding the truss with minimum weight subject to strength constraints under a single loading condition assuming a symmetric structure [Kirsch, 1981]. Allowable stresses are 20000 psi in tension and 15000 psi in compression.
In this classical small size example problem the maximum number of extreme points in a group is 2, whereas it is unity in the ten bar truss and the transmission tower examples above. Indeed, the problem is formulated in terms of two design variables ($n = 2$), on the other hand, at the optimum a single constraint is active ($m' = 1$), which is possible because of the nonlinearity of the stress in the first bar. Therefore, the maximum number of extreme points in a group is $n + 1 - m' = 2$. It has been noted, however, after several runs conducted with different starting points using the $SQPD$ algorithm, that the group of extreme points remains a singleton throughout the solution process. This provides evidence that the maximum number of extreme points in a group is not necessarily reached.
Truss framing Structure. The structure considered in this example is a truss framing subjected to a uniformly distributed vertical load q. Given the problem symmetry, only half the structure is modeled (although one can reduce the problem size further to a quarter). The retained model, as shown

in Figure 4, comprises 380 bars. The truss is to be designed for minimum self weight subject to stress constraints given an allowable stress limit of $250MPa$, and 0.1 cm^2 minimum gage restraint. No variable linking is used in this example. Assuming a load density $q = 5kN/m^2$, an initial design with all cross sections at 2.5 cm^2 is chosen, and using the $SQPD$ algorithm, a final solution is found within 9 SQP iterations. The optimal volume is $13656.4cm^3$. The number of extreme points here again stabilizes at unity at the tail of the sequence and it did not exceed two.
A second run is conducted using a load intensity $q = 10kN/m^2$. The optimal volume of 24397.2 is reached after 17 iterations. However, unlike the previous truss examples the present one converges to a solution with three, that is more than one, extreme points in the final group. The explanation lies again in the nonlinear nature of the problem as illustrated by the three bar truss example. It is also noted that the number of extreme points is 3 at most throughout the sequence of quadratic problems.
Two other runs are conducted using load intensities $q = 20kN/m^2$ (resp. $40kN/m^2$). Convergence is achieved at 30 (resp. 34) iterations. For both runs the optimal group of extreme points counts seven elements. This number, which is not exceeded in the intermediate iterations, is clearly very small in comparison with the problem dimension.

5.1.3 Large Scale Analytical Problems

The test problems considered next are large scale analytical problems. The first two are reference problems $NLP1$ and $NLP2$ that have been treated in [Boggs, Kearsley, Tolle, 1999] to test a penalty type algorithm. Next a parameter dependant family of analytical problems, denoted $NLP3$, is considered.
Problem $NLP1$: This problem is written as follows:

$$NLP1 \quad \left\{ \begin{array}{c} \min 1000[(\sum_{i=1}^{n} x_i^3)^2 - (\sum_{i=1}^{n} x_i^2)(\sum_{i=1}^{n} x_i^4)] \\ \\ x_i - x_{i+1} \leq 0 \;\; i = 1, ..., \; n-1 \\ x_i^2 - x_i x_{i+1}^2 \leq 0 \;\; i = 1, ..., \; n-1 \\ x_1 \geq 0 \;\; x_n \leq 1. \end{array} \right. \qquad (12)$$

It has been solved in [Boggs, Kearsley, Tolle, 1999] for a number of variables $n = 200$. Here, the algorithm $SQPD$ is first applied for n ranging between 200 and 1000. Given that the problem possesses multiple optima, the algorithm is run with different starting points for each value of n. In all cases the number of extreme points at the optimum of each quadratic programming subproblem is less than or equal to four. It should be noted that most of the pivot operations are devoted to the generation of the first extreme point (phase I) which requires in the order of n pivots and often largely exceeds this number. The subsequent extreme points are deduced from the first via only a small number of pivots. It can also be remarked from the data given in Table 4 that, for this example problem, the performance of the modified $SQPD$ algorithm, as measured by the total number pivots, is nearly similar to that of the original algorithm $SQPD$. This happens here because the number of intermediate extreme points and the number of pivots per extreme point generation are so small in his case that no possibility is left for their reduction.

Problem $NLP2$: This problem

$$NLP2 \quad \begin{cases} \min \sum\limits_{i=1}^{n} [10(x_{i+1} - x_i^2)^2 + (1 - x_i)^2] \\ \\ x_i - x_{i+1} \leq 0 \quad i = 1, 3, ..., n-1 \\ 4x_{i+1} - x_i^2 - 4 \leq 0 \quad i = 1, 3, ..., n-1 \\ 2x_{i+1} + x_i - 1 \leq 0 \quad i = 1, 3, ..., n-1 \\ x_1 \geq 0 \quad x_n \geq 0 \end{cases} . \tag{13}$$

is constructed with Rosenbrock's function as an objective. A relevant feature of this function is the tridiagonal structure of the Hessian matrix. Problem (13) has been treated in [Boggs, Kearsley, Tolle, 1999] with $n = 250$. Using the algorithm $SQPD$, the problem has been solved for $n =$ 100, 200, 250, 500 and 750 with corresponding numbers of constraints $m =$ 296, 596, 746, 1496 and 2246, respectively. The number of iterations (nit) to convergence is reported in Table 4. The maximum number of extreme points at the optima of the quadratic programming subproblems is six for all three cases. The master problem size is, therefore, small compared to that of the problem P. Table 4 shows the total number $tnep$ (resp. $tnepm$) of extreme points generated throughout the solution by the algorithm $SQPD$ (resp. modified $SQPD$) and the total number tnp (resp. $tnpm$) of pivot operations performed by the same algorithm. These results demonstrate an advantage of the approximation procedure, that is, the reduction in the number of pivots required for the generation of one extreme point. This reduction can be explained by the specific structure of problem (10) that demands a computational effort generally equivalent to that of a linear system of m equations, whereas problem (7) of the original $SQPD$ method requires the effort of the solution of an arbitrary LP problem.

Problem	n	m	nit	$tnep$	tnp	$tnepm$	$tnpm$
$NLP1$	100	200	16	97	16552	98	16802
	500	1000	19	95	32225	99	31120
	700	1400	17	93	41325	97	42752
	1000	2000	14	82	62552	89	65432
$NLP2$	100	296	60	776	22221	873	14550
	200	596	35	183	14395	180	12630
	250	746	28	211	18792	229	12142
	500	1496	24	190	31384	189	20335
	750	2246	46	446	112160	335	46381

Table 4. Extreme point and pivot count.

Problem $NLP3$**:** A family of example analytical problems are now constructed in the following form:

$$NLP3 \quad \left\{ \begin{array}{l} \min \sum_{i=1}^{n-p} exp((x_i^2-4)(x_i-4)) + \sum_{i=n+1-p}^{n} (x_i^2-1)(x_i-1). \\ \qquad g_i(x) \ = x_i^2 + x_{i+1}^2 - 5 \leq 0, \quad i = 1, ..., n-1 \\ \qquad g_n(x) = x_n^2 + x_{n-1}^2 - 5 \leq 0 \\ \qquad\qquad\qquad\qquad x \geq 0.1 \end{array} \right. \tag{14}$$

where p is a positive integer which controls the number of active constraints at the optimum.
The analytical solution of these problems is trivial. Many example problems have been solved using both the unmodified and the modified $SQPD$

in order to assess the incidence of the initial SEP approach on the computational effort as the problem size increases. The computational load, justifiably measured by the total number of pivots (tnp) involved in the generation of extreme points, is plotted in Figure 5 as a function of the number of variables for $p = 20$ and n ranging from 100 to 1000. The saving achieved by the modified $SQPD$ method is clearly demonstrated. It is noted that the computational advantage improves to greater proportions and reaches 60% as the problem size increases. This is essentially explained by two factors. The first is that the modified method avoids the redundant generation of intermediate extreme points normally carried out in the $SQPD$ algorithm. The second is that the LP subproblems that generate the initial SEP in the modified algorithm are more straightforward to solve than the LP subproblems of the $SQPD$ algorithm because of the a priori knowledge of the basis. The effect of the second factor is clearly illustrated by the case with $n = 450$ where the total number of generated extreme points is nearly the same for both algorithms, whereas the modified algorithm requires only half the number of pivots.

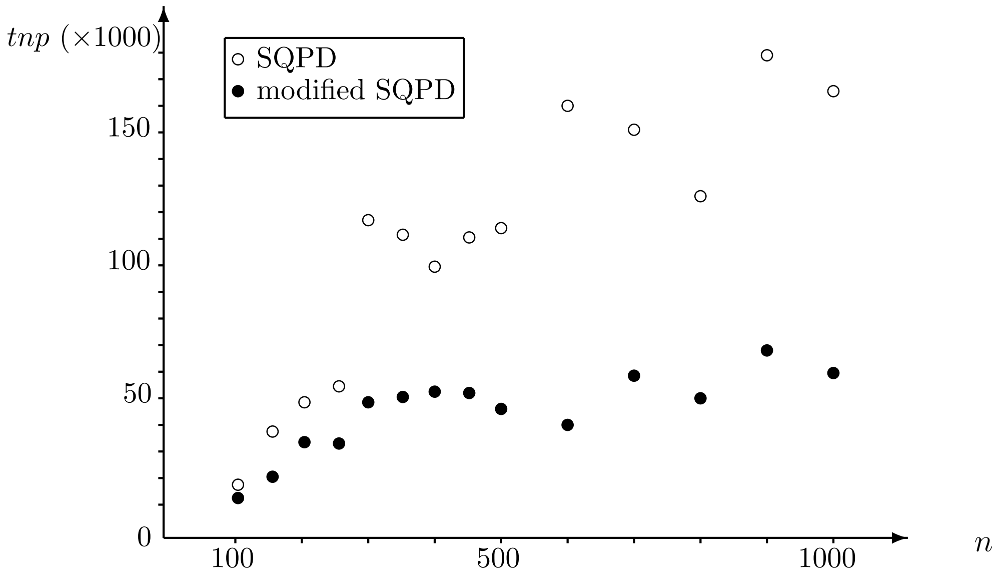

Figure 5. Evolution of total number of pivots versus problem size.

Furthermore, it is noted that, as the problem size is increased the computational effort required by $SQPD$ varies almost linearly, whereas by the modified algorithm it grows in a rather logarithmic pattern. Indeed, in the present example the number $n-m'+1 = p+1$ is constant and the number of extreme points generated by the modified algorithm remains nearly constant independently of n, due to the small number of intermediate extreme points.

5.2 Computational time

All of the numerical examples presented in this paper were run on a 700 Mhz Pentium III personal computer. Computational CPU time in seconds is reported in Table 5 for all runs of the $SQPD$ algorithms and $NLPQL$ code.

6 Conclusion

In the present work Sacher's simple decomposition is applied in solving the quadratic programming problems of the sequence of the SQP algorithm for nonlinear programming. The resulting algorithm naturally preserves the superlinear convergence of the sequential quadratic programming method and provides a capability for handling large scale problems. Moreover, a procedure is developed that aims at reducing the computational effort devoted to the generation of intermediate extreme points. It consists in initiating the decomposition process with a whole set of extreme points, determined from the results of the preceding iteration, without solving a series of master problems and subproblems. Numerical examples of nonlinear programming problems consisting of some structural optimization problems and selected analytical problems are solved using the proposed algorithm in both its original and modified versions and a reference code $NLPQL$. Results indicate improved robustness and excellent accuracy in the solutions obtained by the proposed algorithm. Computational efficiency of the algorithm is particularly high for large problems with a number of active constraints close to the number of variables, such as those encountered in structural optimization. Comparison of the original and the modified versions of the proposed algorithm reveals computational saving up to 60%.

				CPU time(s)		
Problem	n	m	nit	$SQPD$	modified $SQPD$	$NLPQL$
$NLP1$	100	200	16	2,02	2,08	$failed$
	500	1000	19	101,5	100,1	$failed$
	700	1400	17	214,8	217,1	$failed$
	1000	2000	14	532,5	543,3	$failed$
$NLP2$	100	296	60	33,4	20,9	18
	200	596	35	75,5	50,1	111,4
	250	746	28	149,6	103,3	115
	500	1496	24	874,4	549,8	866
	750	2246	46	6030	2732,7	4149
$NLP3$ $p = 20$	100	200	40	4,9	4,3	6,1
	200	400	50	39,1	29,7	64,3
	500	1000	48	232,2	168,9	839,1
	800	1600	54	752,7	531	4811,9
$NLP3$ $p = 10$	100	200	42	2,9	1,8	6,5
	200	400	54	23,4	14,3	68,1
	500	1000	52	235,4	134,1	1051,9
	800	1600	54	558,2	279	5623
Truss design	10	20	6	0,2	0,2	0,3
	27	94	10	1,3	1,3	1,4
	380	760	17	1028	1004	1919

Table 5. Computational time.

References

[1] Ben Daya, M., (1994) A Hybrid Decomposition Approach for Convex Quadratic Programming, King Fahd University of Petroleum and Minerals, under code SE/LINPROG/138.

[2] Biggs, M.C., (1975) Constrained Minimization Using Recursive Quadratic Programming, Towards Global Optimization, L.C.W. Dixon and G.P. Szergo, eds. North Holland, pp.341-349.

[3] Burke, J.V. and Han, S.P., (1989) A Robust Sequential Quadratic Programming Method, Math. Prog., vol.43, pp.277-303.

[4] Boggs, P.T., Kearsley, A.J. and Tolle, J.W., (1999) A Practical Algorithm for General Large-Scale Nonlinear Optimization Problems, SIAM J. Optimization, vol.9, nr.3, pp.755-778.

[5] Boggs, P.T., Kearsley, A.J. and Tolle, J.W., (1999) A Global Convergence Analysis of an Algorithm for Large-Scale Nonlinear Optimization Problems, SIAM J. Optimization, vol.9, nr.4, pp.833-862.

[6] Ciarlet, P.G., (1982) Introduction à l'Analyse Numérique Matricielle et à l'Optimisation, Masson, Paris.

[7] Dantzig, G.B., (1963) Linear Programming and Extensions, Princeton University Press.

[8] Fiacco, A.V. and McCormick, G.P., (1968) Nonlinear Programming : Sequential Unconstrained Minimization Techniques, John Wiley and Sons, NewYork, SEC.2.4.

[9] Han, S.P., (1977) A Globally Convergent Method for Nonlinear Programming, J. Optimization Theory and Applications, vol.22, pp.297-309.

[10] Hearn, D.W., Lawphongpanich, D.W. and Ventura, J.A. (1987) Restricted Simplicial Decomposition: Computation and Extensions, Math. Prog. Study, vol.31, pp.99-118.

[11] Hock, W. and Schittkowski, K., (1981) Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag.

[12] Kirsch, U., (1981) Optimum Structural Design, McGraw-Hill Book Company.

[13] Lachiheb, M., (1997) Extension de la décomposition hybride pour la programmation non linéaire, Mémoire de D.E.A. Mathématiques appliquées, ENIT, Tunis.

[14] Lawrence, C.T. and Tits, A.L., (2001) A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm, SIAM J. Optim., vol.11, nr.4, pp.1092-1118.

[15] Mar'in, A., (1995) Restricted Simplicial Decomposition with Side Constraints, Networks, vol.26, pp.199-215.

[16] Mulvey, J.M., Zenios, S.A. and Ahlfeld, D.P., (1990) Simplicial Decomposition for Convex Generalized Networks, Journal of Information and Optimization Sciences, vol.11, pp.359-387.

[17] Panier, E.R. and Tits, A.L., (1978) A Superlinearly Convergent Feasible Method for the Solution of Inequality Constrained Optimization Problems, SIAM J. Control Optim. vol.25, pp.934-950.

[18] Powell, M.J.D., (1978) Algorithms for Nonlinear Constraints that use Lagrangian Functions, Math. Prog. vol. 14, nr.2.

[19] Powell, M.J.D., (1978) The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations, Nonlinear Programming 3, O.L. Mangasarian R.R. Meyer and S.M. Robinson, eds., Academic Press.

[20] Powell, M.J.D., (1978) A Fast Algorithm for Nonlinear Constrained Optimization Calculations, Numerical Analysis, G.A. Waston ed., Lecture Notes in Mathematics, Springer Verlag, vol. 630, pp.144-157.

[21] Powell, M.J., (1983) On the Quadratic Programming Algorithm of Goldfarb and Idnani, Report DAMTP 1983/Na 19, University of Cambridge, Cambridge.

[22] Rockafellar, R.T., (1970) Convex Analysis, Princeton, New Jersey, Princeton University Press.

[23] Sacher, R.S., (1980) A Decomposition Algorithm for Quadratic Programming, Math. Prog., vol.18, p.16-30.

[24] Schittkowski, K., (1985) NLPQL: A Fortran Subroutine Solving Constrained Nonlinear Programming Problems, Annals of Operations Research, Vol.5, pp.485-500.

[25] Schittkowski K., (2004) NLPQLP20: A Fortran implementation of a sequential quadratic programming algorithm with distributed and non-monotone line search - User's guide, Report, Department of Computer Science, University of Bayreuth.

[26] Schittkowski K., (2005), Optimization in industrial engineering: SQP-methods and applications, Radioss User Meeting, Mecalog, Nice, June 20-22.

[27] Shanno, D.F. and Phua, K.H., (1989) Numerical Experience with Sequential Quadratic Programming Algorithms for Equality Constrained Nonlinear Programming, ACM Transactions on Mathematical Software, vol.15, nr.2, pp.49-63.

[28] Shetty, C.M. and Ben Daya, M., (1988) A Decomposition Procedure for Convex Quadratic Programs, Naval Research Logistics Quarterly, vol.35, pp.111-118.

[29] Ventura, J.A. and Hearn, D.W., (1993) Restricted Simplicial Decomposition for Convex Constrained Problems, Math. Prog., vol.59, pp.71-85.

[30] Von Hohenbalken, B., (1977) Simplicial Decomposition in Nonlinear Programming Algorithms, Math. Prog., vol.13, pp.49-68.

[31] Zillober Ch., Schittkowski K., Moritzen K., (2004) Very large scale optimization by sequential convex programming, Optimization Methods and Software, Vol.18, nr. 1, pp.103-121