Transactions on the Built Environment vol 13, © 1995 WIT Press, www.witpress.com, ISSN 1743-3509

A first order method of moving asymptotes for structural optimization

T. Jiang, P.Y. Papalambros
Design Laboratory, Department of Mechanical Engineering and Applied Mechanics, The University of Michigan, Ann Arbor, MI 48109, USA

Abstract

The method of moving asymptotes (MMA) represents a family of convex approximation methods suitable for structural optimization problems. Its efficiency depends strongly on asymptote and move limit locations. Second order information can be successfully employed for asymptote location but it is too expensive computationally. In this article, two strategies of approximating second order derivatives are derived and implemented in a new algorithm. A selection strategy for move limits to prevent oscillation in the presence of strongly coupled constraints is also described. Computational results for some typical problems are included.

1 Introduction

An important consideration in algorithms for structural optimization is to reduce the number of structural analyses required for objective and constraint function evaluations. To this end, convex approximation methods have been shown to be efficient,[1-4] in part due to extensive function monotonicities in common problem formulations, see, e.g., Nguyen, Strodiot and Fleury.[5]

Several convex approximation methods exist, the first one proposed being the sequential linearization method, see, e.g., Perdersen.[6] Objective and constraint functions are linearized with respect to primary design variables to obtain an approximate LP problem. Another method, proposed by Fleury and Breibant,[1] is the COnvex LINearization method (CONLIN). If the first derivative with respect to a design variable is positive (negative), the function is linearized with respect to the (reciprocal) design variable. Inaccurate approximations cause numerical difficulties. Conservative approximations lead to slow convergence and aggressive ones to oscillation.

The Method of Moving Asymptotes (MMA), developed by Svanberg,[2] uses upper and lower moving asymptotes to adjust the curvature of the approximate functions. Selection of the moving asymptotes is largely heuristic. Smaoui, Fleury and Schmit[3] proposed a selection method using accurate second order information, hereafter referred to as the Second Order Method of Moving Asymptotes (SOMMA). Fleury[4] also proposed a diagonal sequential quadratic programming method. Both methods need accurate second order derivatives.

In this article, a new method for selecting tuning parameters such as asymptotes and move limits in MMA is proposed. In most iterations of the new method, only first derivatives are used to approximate the second order derivatives. Two such approximation strategies are derived that can be combined to address different situations in the choice of asymptotes. A selection strategy for move limits is also proposed that can partially overcome design variable coupling. The resulting algorithm will be referred to as First Order Method of Moving Asymptotes (FOMMA). As in other MMA implementations, the approximate problems are convex and separable. Examples solved by both first and second order methods indicate the efficiency of FOMMA in terms of function calls is higher than SOMMA for most problems, especially for larger problems with strongly coupled constraint functions. An empirical comparison with sequential quadratic programming (SQP) results is included.[6]

2 Convex approximation methods

A mathematical model for structural optimization problems can be written in the following general (negative null) form:

$$\begin{array}{lll} \text{Minimize} & f_0(\mathbf{x}) & \\ \text{subject to} & f_j(\mathbf{x}) \leq 0 & j = 1, 2, \ldots, m \\ \text{and} & LB_i \leq x_i \leq UB_i & i = 1, 2, \ldots, n \end{array} \tag{1}$$

where $\mathbf{x}=(x_1,\ldots,x_n)^T$ is a vector of design variables, $f_0(\mathbf{x})$ is an objective function, typically the structural weight, $f_j(\mathbf{x}) \leq 0$ are behavior constraints, typically limitations on stresses, displacements and buckling, and LB_i and UB_i are given lower and upper bounds on the design variables.

Convex approximation methods generate and solve a sequence of explicit approximate problems until the optimal solution of the original problem is reached. The approximation functions for point $\mathbf{x}^k$ at the k-th iteration in the approximate problems of the MMA algorithm[2] are defined by

$$f_j^k(\mathbf{x}) = r_j^k + \sum_{i=1}^{n} \left[P_{ji}^k/(U_i^k - x_i) + Q_{ji}^k/(L_i^k - x_i) \right] \tag{2}$$

where
$$P_{ji}^k = \begin{cases} (U_i^k - x_i^k)^2 \partial f_j(\mathbf{x}^k)/\partial x_i & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0 \\ 0 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \leq 0 \end{cases} \tag{2a}$$

$$Q_{ji}^k = \begin{cases} 0 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0 \\ -(x_i^k - L_i^k)^2 \partial f_j(\mathbf{x}^k)/\partial x_i & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \leq 0 \end{cases} \tag{2b}$$

$$r_j^k = f_j(\mathbf{x}^k) - \sum_{i=1}^{n} \left[P_{ji}^k/(U_i^k - x_i^k) + Q_{ij}^k/(L_i^k - x_i^k) \right] \tag{2c}$$

$$L_i^k < \alpha_i^k < x_i^k < \beta_i^k < U_i^k \tag{2d}$$

Tuning parameters L_i^k and U_i^k are lower and upper moving asymptotes, and α_i^k and β_i^k are lower and upper move limits respectively. The approximate problem is solved by a dual method. First and second derivatives of the approximate functions are

$$\frac{\partial f_j^k(\mathbf{x})}{\partial x_i} = \begin{cases} P_{ji}^k / (U_i^k - x_i)^2 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0 \\ -Q_{ji}^k / (x_i - L_i^k)^2 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \le 0 \end{cases} \tag{3}$$

$$\frac{\partial^2 f_j^k(\mathbf{x})}{\partial x_i^2} = \begin{cases} 2P_{ji}^k / (U_i^k - x_i)^3 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0 \\ 2Q_{ji}^k / (x_i - L_i^k)^3 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \le 0 \end{cases} \tag{4}$$

$$\partial^2 f_j^k(\mathbf{x}^k) / \partial x_i \partial x_l = 0 \qquad \text{if} \quad i \ne l$$

Note that $f_j^k(\mathbf{x}^k)$ is a first order approximation of $f_j(\mathbf{x}^k)$ at $\mathbf{x}^k$, because

$$f_j^k(\mathbf{x}^k) = f_j(\mathbf{x}^k) \quad \text{and} \quad \partial f_j^k(\mathbf{x}^k)/\partial x_i = \partial f_j(\mathbf{x}^k)/\partial x_i \tag{5}$$

The moving asymptotes L_i^k and U_i^k for the design variable x_i can adjust the curvature of the approximate functions. If asymptotes are selected to provide tight or loose bounds on the variables, the method becomes conservative and slow or aggressive and possibly oscillatory, respectively. For $L_i^k \to -\infty$ and $U_i^k \to +\infty$ for all i, the algorithm becomes sequential linear programming (SLP),[6] while for $L_i^k = 0$ and $U_i^k = +\infty$ for all i, it becomes CONLIN.[1]

In SOMMA, the moving asymptotes are determined so that

$$\partial^2 f_j^k(\mathbf{x}^k)/\partial x_i^2 = \mathrm{Max}[\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2, 0] \tag{6}$$

where $\partial^2 f_j^k(\mathbf{x}^k)/\partial x_i^2$ ($\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2$) are the second order derivatives of approximate (original) functions at point $\mathbf{x}^k$. For i such that $\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2 > 0$, the asymptotes are given by

$$U_{ji}^k = x_i^k + [2\partial f_j(\mathbf{x}^k)/\partial x_i]/[\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2] \qquad \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0$$

$$L_{ji}^k = x_i^k + [2\partial f_j(\mathbf{x}^k)/\partial x_i]/[\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2] \qquad \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \le 0 \tag{7}$$

This concludes the presentation of introductory concepts and nomenclature.

3 Two strategies for approximating second order information

The original MMA is a first order method, a feature desirable to preserve while attempting to increase approximation accuracy. MMA guarantees that both function and derivative values of the approximate and original functions are equal at the approximation point $\mathbf{x}^k$.

Strategy 1 involves making the first derivatives of the approximate functions equal the derivatives of the original functions at $\mathbf{x}^k$ *and* $\mathbf{x}^{k-1}$ at the kth iteration.[8] From Eqn(3), the moving asymptotes are then determined so that

$$\frac{\partial f_j(\mathbf{x}^{k-1})}{\partial x_i} = \begin{cases} P_{ji}^k / (U_i^k - x_i^{k-1})^2 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0 \\ -Q_{ji}^k / (x_i^{k-1} - L_i^k)^2 & \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \le 0 \end{cases} \tag{8}$$

The right hand side are the derivatives of the approximate functions at $\mathbf{x}^{k-1}$

Substitution of P_{ji}^k and Q_{ji}^k from Eqn(2) into Eqn(8) yields

$$U_{ji}^k = x_i^k + (x_i^k - x_i^{k-1}) / (c_{ji}^{1/2} - 1) \text{ and } L_{ji}^k = x_i^k + (x_i^k - x_i^{k-1}) / (c_{ji}^{1/2} - 1) \quad (9)$$

where $c_{ji} = [\partial f_j(\mathbf{x}^k)/\partial x_i^k] / [\partial f_j(\mathbf{x}^{k-1})/\partial x_i^{k-1}]$ (10)

Strategy 2 computes approximate second derivatives instead of accurate second derivatives using quasi-Newton updates. The difference of the derivatives between the k and k-1 iteration points can be used. The approximate second derivatives of $f_j(\mathbf{x}^k)$ at $\mathbf{x}^k$ are then

$$H_{ji}^k = \frac{[\partial f_j(\mathbf{x}^k)/\partial x_i - \partial f_j(\mathbf{x}^{k-1})/\partial x_i](x_i^k - x_i^{k-1})}{\|x_i^k - x_i^{k-1}\|^2} \approx \frac{\partial^2 f_j(\mathbf{x}^k)}{\partial x_i^2} \quad (11)$$

Moving asymptotes are computed from Eqn(7) using H_{ji}^k instead of $\partial^2 f_j(\mathbf{x}^k)/\partial x_i^2$:

$$U_{ji}^k = x_i^k + [\partial f_j(\mathbf{x}^k)/\partial x_i] / H_{ji}^k \qquad \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i > 0$$
$$L_{ji}^k = x_i^k + [\partial f_j(\mathbf{x}^k)/\partial x_i] / H_{ji}^k \qquad \text{if} \quad \partial f_j(\mathbf{x}^k)/\partial x_i \le 0 \quad (12)$$

The asymptotes now depend not only on the corresponding variables x_i but also on the objective and constraint functions. A line search is now required to compute the dual objective function, a disadvantage of this approach.

Tuning of the MMA algorithm requires use of a different strategy at each iteration depending on whether the resulting approximate problem is desired to be conservative or aggressive. Since both L_{ji}^k and U_{ji}^k in Eqn(9) and (12) are of similar form, for simplicity only U_{ji}^k will be used to illustrate the situation. Also, for clarity, since the approximation problem is separable, only the one dimensional case will be examined. For Strategy 2,

$$H_{ji}^k = [\partial f_j(\mathbf{x}^k)/\partial x_i - \partial f_j(\mathbf{x}^{k-1})/\partial x_i] / (x_i^k - x_i^{k-1}) \quad (13)$$

and Eqn(9) and Eqn(12) are rewritten as

$$\text{Strategy 1:} \quad U_{ji}^k = x_i^k + C_1(x_i^k - x_i^{k-1}) \quad (14)$$

$$\text{Strategy 2:} \quad U_{ji}^k = x_i^k + C_2(x_i^k - x_i^{k-1}) \quad (15)$$

where $C_1 = 1/(c_{ji}^{1/2} - 1)$ and $C_2 = 2c_{ji} / (c_{ji} - 1)$ (16)

The coefficients C_1 and C_2 determine how far the moving asymptotes are from $\mathbf{x}^k$ and how conservative or aggressive the approximations are. The larger they are, the farther the moving asymptotes are placed, and the more aggressive is the method. As illustrated in Figure 1, when c_{ji} is bigger than one Strategy 2 is more aggressive than Strategy 1, the opposite being true when c_{ji} is less than one. To stabilize the algorithm, a conservative decision should be made, i.e., for $c_{ji} > 1$ Strategy 1 is used, and for $c_{ji} < 1$ Strategy 2 is used. From Eqn(10), when $c_{ji} > 1$ the absolute value of the first derivative at the current iteration point increases, and when $c_{ji} < 1$ it decreases. Since the approximate functions are monotonic with respect to x_i and the problem is in negative null form, when the absolute values of the first derivatives at iteration points increase, future points will tend to become infeasible. Similarly, when the absolute values decrease, future points will tend to be feasible. Strategy 1 is used in the first case and Strategy 2 in the second.

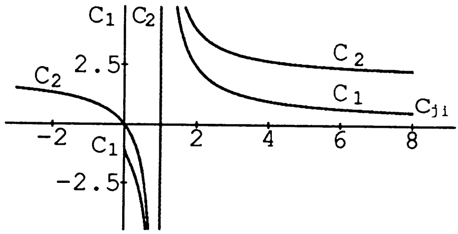

Figure 1: Coefficients of moving asymptotes.

Since the approximate functions are monotonic and the FOMMA algorithm needs derivatives at the previous point to approximate the second order derivatives at the current point, there are some special cases which need to be dealt with. First, if the first derivative of the original function at a point is zero, a convex approximation method cannot approximate this point because the approximate problem is monotonic. From Eqn(10) and Eqn(14)-(16), if $\partial f_j(\mathbf{x}^{k-1})/\partial x_i^{k-1}) \to 0$ then $C_1 \to 0$ and $U_{ji}^k \to x_i^k$. If $\partial f_j(\mathbf{x}^k)/\partial x_i^k \to 0$, then $C_2 \to 0$ and $U_{ji}^k \to x_i^k$. In this case, the algorithm fails to find a new point $\mathbf{x}.^{k+1}$ To remedy this, a linear approximation can be used to select the moving asymptotes, i.e., U_{ji}^k is set equal to a big number.

In general, the derivatives of the objective and active constraint functions at an optimum will not be zero, unless an unconstrained optimum and/or degenerate activity exists. Optima in engineering problems are usually on the boundary of the feasible domain.[9] However, because the approximate functions are monotonic, the MMA algorithms fail to solve unconstrained problems. If the derivative of the constraint function at an optimum is zero, either the feasible domain in this dimension is a point instead of a set or this point is an inflection point. These cases seldom happen in engineering problems. Should such a case occur, linear approximations can be used as mentioned previously.

Second, if the sign of a first order derivative with respect to a design variable changes between two consecutive iteration points, the function is not monotonic in that segment with respect to the design variable or the function is coupled among the design variables. Strategy 1 cannot be used as it will violate the square root definition in Eqn(9). Only Strategy 2 can be used in this case.

Third, the approximate problem should be convex, otherwise a dual gap will exist and the dual solution may not be the solution to the primal approximate problem. This happens if the approximate second order derivative H_{ji}^k is negative. In order to guarantee convexity of approximation functions, a linear approximation can be used, setting U_{ji}^k to a big number.

Fourth, a design variable may not vary after an iteration, i.e., $x_i^k = x_i^{k-1}$. This case happens often when the optimum is at a variable bound. From Eqn(11), if $x_i^k = x_i^{k-1}$ then $H_{ji}^k = 0$, but actually at this point the function probably is not linear. In this case, the previous moving asymptotes U_{ji}^{k-1} are used for the kth iteration.

Finally, since at the first iteration the above strategies cannot be used, the

Transactions on the Built Environment vol 13, © 1995 WIT Press, www.witpress.com, ISSN 1743-3509

second derivatives are calculated using the finite difference method.

4 Selection of move limits

Move limits are essentially tuning parameters for global convergence. The strategies of the previous section balance conservative and aggressive decisions and more accurately approximate objective and constraint functions, so it is not necessary to select move limits close to $\mathbf{x}.^k$ Move limits can be usually selected as

$$\alpha_i^k = \mathrm{Max}\{BU_i,\ 0.99U_{ji}^k + 0.01x_i^k \ \mid\ U_{ji}^k < x_i^k\}$$
$$\beta_i^k = \mathrm{Min}\{BL_i,\ 0.99U_{ji}^k + 0.01x_i^k \ \mid\ U_{ji}^k > x_i^k\} \quad (17)$$

and will change as the asymptotes move.

As mentioned previously, a difficulty arises when an active constraint is not monotonic with respect to a design variable or coupled among design variables. The derivative of the constraint function with respect to such a design variable may change sign at each consecutive iteration. MMA cannot give a good approximation in this case and iterations will oscillate. The algorithm becomes extremely slow. To address this situation, if the derivative sign of a function with respect to x_i changes at the kth iteration, move limits are added at the middle point between x_i^k and x_i^{k-1}, i.e.,

$$A_i^k = (x_i^k+x_i^{k-1})/2 \qquad \text{if} \qquad x_i^k > x_i^{k-1}$$
$$B_i^k = (x_i^k)+x_i^{k-1})/2 \qquad \text{if} \qquad x_i^k < x_i^{k-1} \quad (18)$$

Although this is not an accurate selection, it was found to work well in practice.

5 Algorithm for moving asymptotes location

The strategies described above for determining the asymptotes are now stated more formally for the U_{ji}^k values. At the kth iteration, for i = 1,..., n and j= 1,..., m, the algorthm proceed as follows.

(0) Select parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3, M$.

(1) At k = 0, let $H_{ji}^k = \partial^2 f_j(\mathbf{x}^k)/\partial x_i^2$; if $H_{ji}^k \le 0$ then set $H_{ji}^k = \varepsilon_1$ and go to Step 7.

(2) At k > 0, if $|\mathbf{x}^k - \mathbf{x}^{k-1}| < \varepsilon_2$ go to Step 10; otherwise continue.

(3) Compute H_{ji}^k from Eqn(11); if $H_{ji}^k \le 0$ then set $H_{ji}^k = \varepsilon_1$ and go to Step 7.

(4) If $\partial f_j(\mathbf{x}^k)/\partial x_i\ \partial f_j(\mathbf{x}^{k-1})/\partial x_i \le 0$ compute A_i^k and B_i^k from Eqn(18); otherwise set $A_i^k(B_i^k)$ to a large negative (positive) number and go to Step 7.

(5) If $|\partial f_j(\mathbf{x}^k)/\partial x_i| \le |\partial f_j(\mathbf{x}^{k-1})/\partial x_i|$ go to Step 7; otherwise continue.

(6) [Strategy 1] Compute asymptotes from Eqn(9). Go to 8.

(7) [Strategy 2] Compute asymptotes from Eqn(12).

(8) If $\partial f_j(\mathbf{x}^k)/\partial x_i \ge 0$ and $U_{ji}^k - x_i^k < \varepsilon_3 x_i^k$ or $U_{ji}^k - x_i^k > Mx_i^k$ then set $U_{ji}^k = x_i^k + Mx_i^k$.

(9) If $\partial f_j(\mathbf{x}^k)/\partial x_i < 0$ and $U_{ji}^k - x_i^k > -\varepsilon_3 x_i^k$ or $U_{ji}^k - x_i^k < -Mx_i^k$, then set $U_{ji}^k = x_i^k - Mx_i^k$

Transactions on the Built Environment vol 13, © 1995 WIT Press, www.witpress.com, ISSN 1743-3509

(10) Set $\alpha_i^k = \max\{BL_i,\ A_i^k, 0.99U_{ji}^k + 0.01x_i^k \mid U_{ji}^k < x_i^k)\}$

$\beta_i^k = \min\{BU_i,\ B_i^k, 0.99U_{ji}^k + 0.01x_i^k \mid U_{ji}^k > x_i^k\}$

Some remarks on the above steps would be useful . For Step 0, the numbers $\varepsilon_1 = 10^{-12}$, $\varepsilon_2 = 10^{-12}$, $\varepsilon_3 = 10^{-2}$, $M = 10^3$ usually work well. They are used to scale the approximate problem. At Step 1, the first iteration, accurate second derivatives are used. If one of them is non-positive, the function is locally linear or concave with respect to the design variable. In this case, a small value is assigned to H_{ji}^k, and U_{ji}^k will be far from x_i^k and a linearization is effected. Step 2 means that if a variable does not vary at an iteration, the asymptotes at that iteration are not changed. Step 3 uses Strategy 2 and indicates again a linarization. Step 4 indicates the heuristic used when signs of the first derivatives of the original functions change after an iteration, and move limits are posed at the mid point of (x_i^k, x_i^{k-1}). In Step 5 the derivatives decrease, and Strategy 2 will be used. Step 8 is executed, if the derivative of an original function is greater then zero, $U_{ji}^k > x_i^k$. Then $U_{ji}^k - x_i^k < \varepsilon_3 x_i^k$ means that $\partial f_j(x^{k-1})/\partial x_i^{k-1}$ is close to zero in Strategy 1 or $\partial f_j(x^k)/\partial x_i^k$ is close to zero in Strategy 2. The asymptotes will be very close to x_i^k in either case and linear approximations are used, Mx_i^k being a big number compared to x_i^k. Similarly, $U_{ji}^k - x_i^k > Mx_i^k$ means that the asymptotes are too far from x_i^k. If $U_{ji}^k \rightarrow \infty$, then The dual objective function of the approximation problem will go to infinit. Therefore, $U_{ji}^k - x_i^k$ is limited to less than Mx_i^k. A similar situation but with negative derivative is addressed in Step 9. Step 10 determines move limits based on Eqn(17) and (18).

6 Test examples

In this section, two test examples are presented. In order to compare the function calls among FOMMA, SOMMA, original MMA and SQP, it is assumed that all methods use the finite difference method to compute first and second derivatives. Both FOMMA and MMA require one objective and constraint function call and one first order derivative in each iteration. SOMMA, in addition, requires one second order derivative in each iteration. Function calls for SQP vary among different iterations.

6.1 Cantilever beam problem[2]

The weight of a cantilever beam (Figure 2) built from five square box beam elements with uniform pre-arranged wall thicknesses is minimized subject to a constraint on the vertical tip displacement under a vertical load applied at the tip. Design variables are the sizes of the five square boxes. Objective values at each iteration and total function calls (last row) are tabulated in Table 1. The results of the best original MMA come from Svanberg[2] (based on empirical selection of moving asymptotes values). SQP required most iterations and function calls.

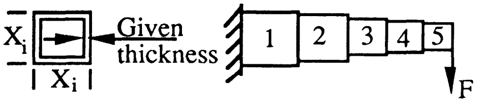

Figure 2: Cantilever beam.

Table 1 The iteration history of the cantilever beam problem

Iterations	FOMMA	SOMMA	MMA	SQP
0	1.560	1.560	1.560	1.560
1	1.378	1.378	1.309	1.552
2	1.347	1.342	1.335	1.511
3	1.338	1.340	1.340	1.348
4	1.340			1.393
5				1.364
6				1.319
7				1.338
8				1.346
9				1.344
10				1.340
f_calls	29	33	18	66

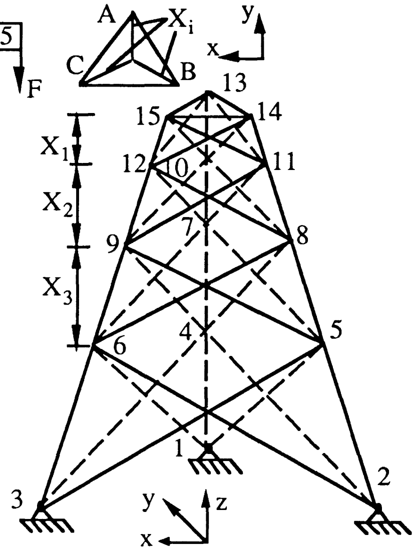

Figure 3: 39 bar problem.

6.2 39-bar tower problem[10]

The cantilever bean problem is too simple to test efficiency. A 39-bar tower (Figure 3) is tested. Weight is minimized subject to stress, buckling and a single deflection constraints under three loads applied at nodes 13, 14 and 15 in the positive y-direction. There are six configuration and five size variables (areas of the five groups of elements). The configuration variables are distances between different levels of the tower and distances from the central point to the three corner points of the tower at different levels.

Two cases are solved. In Case I, a single deflection constraint at node 14 or 15 in the positive y-direction and 26 stress constraints are considered. The initial point is feasible. In Case II, buckling constraints are added to all elements. Total number of constraints is 53. Initial point is the optimum of Case I, and it is infeasible.

Table 2 The iterations of 39-bar problem

	Case I			Case II		
	FOMMA	SOMMA	SQP	FOMMA	SOMMA	SQP
Iterations	17	26	233	12	21	101

Iterations numbers are shown in Table 2. In Case I, up to the thirteenth iteration, SOMMA converges faster than FOMMA with respect to the number of iterations, since FOMMA employs conservative approximations. Close to the optimum FOMMA converges faster as it uses information at two points instead

Transactions on the Built Environment vol 13, © 1995 WIT Press, www.witpress.com, ISSN 1743-3509

of one to approximate the functions. The more "global" approximation in FOMMA will tend to avoid oscillation near the optimum. SQP requires many more iterations and function calls than both FOMMA and SOMMA. Results for Case II present the same trends as Case I.

7 Conclusion

The proposed MMA variant uses two strategies to approximate second order derivatives and modulate how agressive the search may be. The algorithm is easy to use since all tuning parameters are automatically chosen. Test examples confirm that the algorithm is expected to be more efficient than both SOMMA and SQP, especially for larger problems.

References

1. Fleury, C. & Breibant, V. Structural optimization : a new dual method using mixed variables, *International Journal of Numerical Method in Engineering*, 1986, **23**, 409-428.
2. Svanberg, K. The method of moving asymptotes a new method for structural optimization. *International Journal of Numerical Methods in Engineering*, 1987, **24**, 359-373.
3. Smaoui, H., Fleury, C. & Schmit, L.A. Advances in dual algorithms convex approximation methods. In Proceedings of Proceeding of AIAA/ASME/ASCE 29th Structural Dynamic and Materials Conference, Williamsburg.VA. 1988, 1339-1347.
4. Fleury, C. Efficient approximation concepts using second order information, *International Journal for Numerical Methods in Engineering*, 1989, **28**, 2041-2058.
5. Nguyen, V.H., Strodiot, J.J. & Fleury, C. A mathematical convergence analysis of the convex linearization method for engineering design optimization. *Engineering Optimization*, 1987, **11**, 195-216.
6. Pedersen, P. The integrated approach of FEM-SLP for solving problems of optimal design. In: *Optimization of distributed parameter structures*, Edited by Huang, E. J. and. Cea, J., Sijthoff &Noordhoff, 1981, 757-780.
7. Schittkowski K. NLPQL: A FORTRAN subrouting solving constrained nonlinear programming problems, *Annals of Operations Research*, 1985, **5** (6), 485-500.
8. Haftka, R.K., Nachlas, J.A. & Watson, L.T. Two-point constraint approximation in structural optimization, *Computer Methods in Applied Mechanics and Engineering*, 1987, **60**, 289-301.
9. Azarm, S. & Papalambros, P., An Automated Procedure for Local Monotonicity Anaysis, *trans. ASME J. of Mechanisms, Transmissions, and Automation in Design*, 1984, **106**, 82-89.
10. Bremicker M., Papalambros, P.Y. & Loh H.T. A sequential linearization approach for solving mixed-discrete nonlinear design optimization problems with a new sequential linearization algorithm. *Compu.&Struct.*, 1990, **37**, 451-461.