SUITABILITY OF OPTIMIZATION PACKAGES FOR AN MDO ENVIRONMENT

D. HAIM, A. A. GIUNTA, M. M. HOLZWARTH
W. H. MASON, L. T. WATSON

Multidisciplinary Analysis and Design Center for Advanced Vehicles
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0203

AND

R. T. HAFTKA

Department of Aerospace Engineering, Mechanics and Engineering Science
University of Florida
Gainesville, Florida 32611-6250

ABSTRACT

An examination of the performance of several optimization packages and their suitability for inclusion in a realistic multidisciplinary design optimization (MDO) environment is conducted. The packages are incorporated into a High-Speed Civil Transport (HSCT) aircraft design code and are used with both a response surface (RS) model and a more detailed, noisy model. While most packages converge to similar designs, there are large variations in CPU times and usability. Results are reported for a SGI workstation.

KEY WORDS Aircraft design Constrained optimization Multidisciplinary design Optimization Software

INTRODUCTION

For future aircraft designs to continue to improve, more systematic approaches to the integration of the key flight vehicle disciplines must be developed. Traditional design approaches are reaching their limits, and multidisciplinary design optimization (MDO) methodology is being developed in response to this need. While single discipline optimization is important, in developing a complete system approaching an optimum, many disciplines must be incorporated simultaneously. For example, the thickness of the wing has a profound influence on both the weight of the wing, where thicker is better, and the aerodynamic drag, where thinner is better. The system cannot be optimized unless both disciplines are considered together. Currently the combined optimization is done early in the process using simple algebraic models for the disciplines. Once a key configuration design variable, i.e., the wing thickness, is selected, each discipline then works independently with much more precise methods using the selected design variable as an equality constraint. To consider many disciplines simultaneously using very precise numerical methods is an enormously expensive computational effort. An optimization procedure might like to have hundreds, if not thousands, of evaluations of the structures and aerodynamics involving massive finite element models of the structures and Navier-Stokes solutions for the aerodynamics.

The overall goal is to allow the MDO of a High Speed Civil Transport (HSCT) to be carried out so that the accuracy of the computationally expensive methods can influence the design immediately, before key design variables are fixed. The focus of the present work is on the formulation of the optimization problem and not the optimization algorithm and software. Thus, one of the important considerations in the work is the selection of the optimization package. The optimization package must require the fewest possible function evaluations while coping with a wide variety of engineering analysis results. Often the engineering results do not vary smoothly with respect to design variable changes. The software engineering considerations associated with ease of use and flexibility are also important. This paper presents the results of studies intended to assess several optimization packages for use in a MDO context.

HSCT OPTIMIZATION PROBLEM

For testing purposes a simple five variable HSCT wing design problem is used here. This five variable problem is a subset of the "full" HSCT design problem which employs twenty-nine design variables to describe an HSCT wing/fuselage/tail/engine configuration.

The objective of the five variable HSCT wing design problem is to minimize the takeoff gross weight (TOGW) of the HSCT configuration within the allowable limits of the design variables. These variables are root chord (C_{root}), tip chord (C_{tip}), inboard leading-edge sweep angle (Λ_{LE}), thickness-to-chord (t/c) ratio, and fuel weight (W_{fuel}), which had initial values of $C_{root} = 185.0$ ft, $C_{tip} = 10.0$ ft, $\Lambda_{LE} = 75.0°$, t/c ratio $=$ 2.0%, and $W_{fuel} = 315,000$ lb. The wing planform and airfoil section definitions are shown in Figure 1.

To uniquely define the wing planform from the four wing variables, the length of the leading-edge from the wing/fuselage junction to the leading-edge break is held constant as is the outboard leading-edge sweep angle. For the airfoil section definitions, the leading-edge radius is held constant along with the chord-wise location of maximum thickness.

In the five variable example problem the fuselage definition is held fixed with a fuselage length of 300 ft and an internal volume of 23,720 ft^3. Additionally, the vertical tail area is held fixed at 700 ft^2, and the engine thrust is held constant at 39,000 lb per engine. A horizontal tail is not considered for this aircraft. The mission profile is simplified from the "full" HSCT aircraft design problem to include only a supersonic cruise leg and landing. The altitude for the Mach 2.4 supersonic cruise mission is held constant at 65,000 ft. Landing constraints on both the overall lift coefficient (C_L) and eighteen wing section lift coefficients (C_l) were examined for emergency landing situations. The forty-two nonlinear constraints for this problem consist

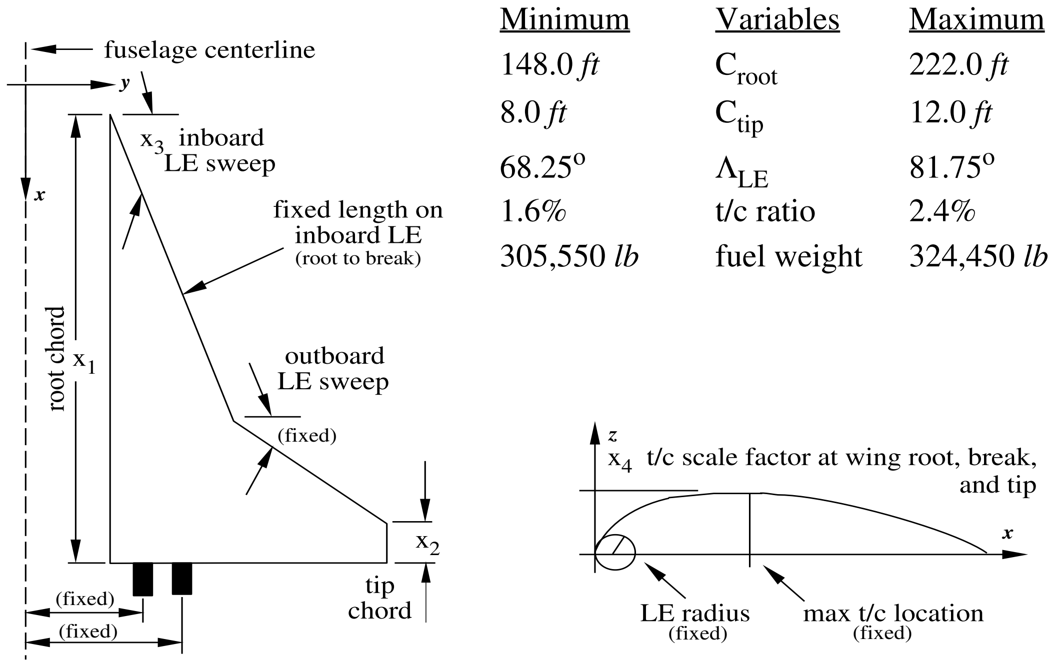

Figure 1. The HSCT wing design problem.

of both geometric constraints (e.g., all wing chords $\geq$ 7.0 ft), and aerodynamic/performance constraints (e.g., C_L at landing ≤ 1, and range $\geq$ 5,500 n.mi.). The five variable HSCT wing design optimization problem may be expressed as

$$\min_{x \in R^5} \mathrm{TOGW}(x), \text{ subject to } g_i(x) \leq 0, \qquad i = 1, \ldots, 42, \tag{1}$$

where x is the five dimensional vector of design variables, and g is the forty two dimensional vector of nonlinear inequality constraints[1].

MDO PARADIGM

This section reviews the HSCT design process[2]. Response surface (RS) methodology is a statistical technique in which smooth functions (*response surfaces*), typically polynomials, are used to model an objective function or constraints. For example, a quadratic response surface model (also known as a regression model) has the form

$$y = c_0 + \sum_i c_i x_i + \sum_{1 \leq i \leq j \leq p} c_{ij} x_i x_j + \epsilon, \tag{2}$$

where the x_i are the design variables, the c_i are the polynomial coefficients, y is the measured response, and ϵ is a random error term. Typically, the polynomial (regression) coefficients are estimated using the method of least squares.

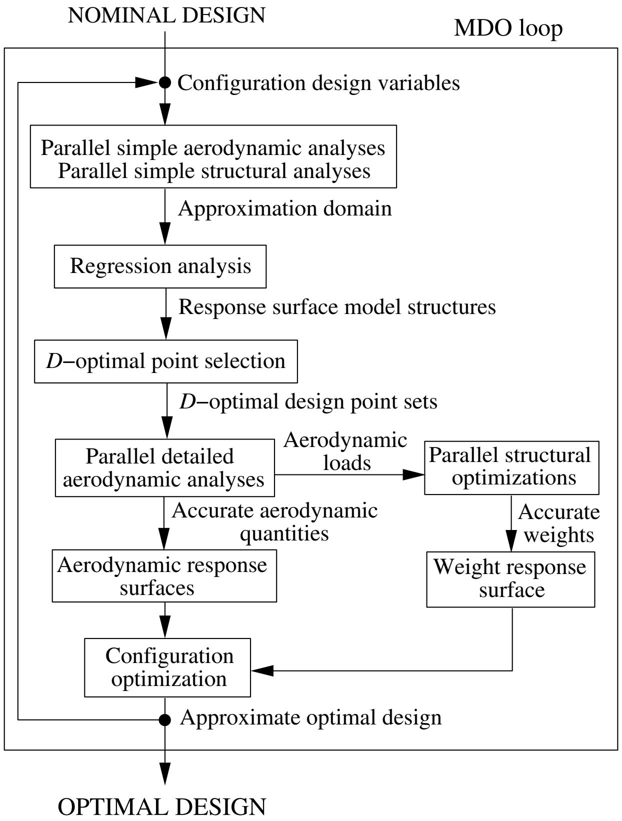

Figure 2. MDO Paradigm.

The MDO algorithm is an iterative process, with a nominal design as its starting point, approximate optimal designs as loop iterates, and an optimal design as its ending point (Figure 2). At the start of each loop, aerodynamic shape and mission variables are obtained from either the nominal starting design or the intermediate approximate optimal design. These shape and mission variables are then used in the parallel simple aerodynamic and structural analyses.

The simple aerodynamic analyses are performed on a regular grid of points in the design space. Simple aerodynamic calculations evaluate the (aerodynamic) feasibility of each grid point using tolerances on the constraints and move limits on the objective function, eliminating grossly infeasible points, and generating an approximation domain. The simple structural analyses use the aerodynamic shape and mission variables in basic weight equations to calculate approximate weights needed by the objective function and constraints, further refining the approximation domain.

Using the relatively abundant data from the simple analyses, regression analysis and analysis of variance are used to identify less important terms in the polynomial response surface models. Once the less important terms are eliminated, the structure of the reduced-term polynomial regression models is known, and can be used later in the generation of response surface approximations of the optimal weight and necessary aerodynamic quantities over the approximation domain. The structure of a response surface model is embodied in the matrix $\mathbf{X}$, which defines the merit function $|\mathbf{X}^T\mathbf{X}| = \det \mathbf{X}^T\mathbf{X}$.

A genetic algorithm and, more recently, an exchange algorithm, are used to find sets of approximate D-optimal[3] design points in the approximation domain obtained from the parallel simple analyses. A subset of m points in the approximation domain is called D-optimal for a particular regression model if that subset maximizes the merit function $|\mathbf{X}^T\mathbf{X}|$ over all possible subsets of m points from the approximation domain. These D-optimal design points are given as input to the detailed aerodynamic analysis code, which performs detailed analyses at each of the D-optimal design points in parallel. The analyses result in accurate aerodynamic quantities, such as wave drag and other drag components, and accurate aerodynamic loads.

The accurate aerodynamic quantities are used to generate reduced-term polynomial response surface models for each of the expensive quantities (such as wave drag). An aerodynamic load calculated in the detailed aerodynamic analyses is used in a detailed structural optimization to calculate an accurate optimal weight for that particular aerodynamic load. This structural optimization is done (in parallel) for each aerodynamic load generated in the detailed aerodynamic analyses. The accurate optimal weights calculated in the structural optimization are used to generate a reduced-term polynomial response surface model for the optimal weight.

All the response surface models are then fixed for all subsequent iterations, and are used in a configuration optimization to generate an approximate optimal design, which will be used as the starting design for the next iteration of the MDO loop. The grid spacing may possibly be refined for the simple analyses. When some convergence criterion is satisfied, the MDO loop exits with an optimal design.

OPTIMIZATION PACKAGES

This section reviews briefly the optimization packages considered. All the packages are written in FORTRAN77. Detailed information is provided in the cited references.

DOT

Design Optimization Tools (DOT)[4] is a commercial package for general unconstrained and constrained optimization problems. The formal problem statement for DOT includes only nonlinear inequality constraints. One can form an equality constraint by defining two separate inequality constraints.

DOT offers different methods of optimization for unconstrained and constrained problems. In the unconstrained case the user has a choice between quasi-Newton search directions using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula for Hessian updates, or a Fletcher-Reeves (FR) conjugate gradient method for the search directions. In the constrained case the user has a choice between a modified method of feasible directions (MFD), where the search direction is determined by critical constraints, sequential linear programming, or sequential quadratic programming (SQP). Both MFD and SQP methods were tested for the HSCT problem in this study.

FFSQP

FORTRAN Feasible Sequential Quadratic Programming (FFSQP)[5] is a code for nonlinearly constrained optimization handling both equality and inequality constraints. As in DOT, nonlinear equality constraints are turned into inequality constraints. FFSQP uses the BFGS formula with Powell's modification for Hessian updates. It operates in the feasible region and, therefore, tests if the starting point is feasible. If the initial design is infeasible, FFSQP first generates a feasible starting point. It then modifies the objective function to include penalty parameters that are iteratively adjusted. Two algorithms, from which the user can choose, are implemented for decreasing the objective function value. In the first mode the modified objective function decreases at each iteration after feasibility has been reached. The method used is an Armijo-type line search with monotone decrease of the objective function at each iteration. The second mode uses a non-monotone line search, forcing the decrease of the objective function within, at most, four iterations.

LANCELOT

LANCELOT[6] is a package for large-scale nonlinear optimization. The emphasis in LANCELOT is on problems which are significantly nonlinear, that involve a large number of nonlinear equality constraints. The authors provided a simplified driver tailored to small unstructured problems like the HSCT design. The simple driver offers, like the original version, methods for optimization with only bound constraints and equality constraints.

Like SQP, LANCELOT builds a quadratic approximation for a nonlinear objective function, which is then incorporated into an augmented Lagrangian objective function with the constraints. LANCELOT finds an approximation to a generalized Cauchy point in the intersection of the feasible box and the trust region, which is a region of the design space where the quadratic approximation is indeed valid. This Cauchy point is then used in the approximate minimization of the quadratic model. LANCELOT does not deal well with inequality constraints handled via slack variables.

Table 1 Implementation and usage of the optimization packages

	Studying	Coding	Installing	Usage	Speed	Documentation	Default Output
DOT	1.5h	1.5-2h	10m	easy	very fast	excellent	very informative
FFSQP	1h	45m	10m	easy	very slow	moderate	non informative
LANCELOT	2h	2.5h	1h	easy	slow	excellent	very informative
NEWSUMT-A	1.5h	2.5h	10m	easy	moderately slow	moderate	very informative

NEWSUMT-A

NEWSUMT-A[7] is a modern version of the classic sequential unconstrained minimization technique (SUMT)[8] for the solution of linear and nonlinear constrained optimization problems. In SUMT a constrained problem is transformed into a sequence of unconstrained problems. In the unconstrained objective function an extended interior penalty function is used for inequality constraints, and an exterior penalty function for equality constraints. Newton's method with approximate second derivatives is used in the unconstrained optimization to find the search direction. If first derivatives are not available, the gradients are estimated by finite differences.

IMPLEMENTATION ISSUES

This section compares the implementation and performance of each of the above packages. The next section discusses in detail the convergence of each package.

Table 1 shows the times (based on careful records kept during the process) for the various steps involved in implementing the optimization packages. The first column, studying, is the time involved in reading the manuals and studying the various parameters which are required to mesh the package with the HSCT MDO code. The second column, coding, shows the actual programming time invested in incorporating the package into the HSCT code. It also includes the time for testing the integrated codes. The third column, installing, shows the time spent in obtaining the packages and tailoring them to the SGI architecture. Apart from LANCELOT, all the packages were installed within ten minutes. LANCELOT involves many subprocedures and needs to be tailored to the architecture on which it will run. This process takes about an hour, but no major complications were encountered. However, LANCELOT seems to be more sensitive to the machine environment than the other packages. LANCELOT converged to different points on a SGI Indigo^2 and a SGI Power Challenge, whereas all the other codes converged to identical points on the two SGI machines.

With regard to speed, the relative performance is reported for runs which use the response surface models. DOT-SQP is very fast, however, DOT-MFD is only moderately fast. LANCELOT, on the other hand, is slow for equality constraints and very slow for inequality constraints with slack variables. As for the intermediate output, FFSQP in its default mode is non-informative, and is otherwise moderately informative.

RESULTS

Each package was started from ten different points of the design space, five were feasible and five infeasible. To the extent possible, the convergence criteria were comparable for all packages. For LANCELOT convergence occurs when the norms of both the projected gradient and the equality constraints are less than 10^{-4}. The same value was used for the final norm requirement for the Newton direction in FFSQP. For both DOT and NEWSUMT-A convergence was declared when the difference in the objective function for two successive iterations was less than 10^{-4}.

Table 2 presents the results for optimizations using response surface approximations in the HSCT code. The top number in each block represents the total CPU time in seconds, excluding the time it takes to create the response surfaces (4–5 CPU hours), on a 100MHz R4000SC SGI Indigo^2 processor. The middle number represents the number of iterations needed to reach convergence, and the bottom number represents the final value of the objective function TOGW in lbs.

DOT-SQP is clearly the fastest method. Lancelot was the only package which failed to converge properly in two out of the ten starting points. For the other eight points it reached gross weights which were about 0.05% lower than those from the other programs. However, after examining the Lagrange multipliers at several optima, it was concluded that these better optima are entirely due to Lancelot converging with constraint violations, which reduced the objective function (TOGW) by 0.05% more than that permitted by the constraint violations of other packages. The CPU time factor between the above two packages is between 6 and 22 in favor of DOT-SQP. NEWSUMT-A converges, on average, within the same time scale as LANCELOT; however, its optima resemble those of DOT-SQP. FFSQP has the longest convergence time, by a factor of up to 140 worse than DOT-SQP, and furthermore converges to the worst optima in most cases. FFSQP converges for all starting points to within one lb. of the same optimum value.

The above results should be viewed with caution. For the HSCT design all of the above optima lead to similar designs. The maximum difference of 0.08% in the optimum design (515 lbs. maximum) is not significant for HSCT design. Thus, DOT-SQP seems to be the clear favorite.

Table 3 shows the equivalent results using a detailed model in the HSCT code, i.e., all function values are computed directly from HSCT analysis codes. Unlike the RS approach, the analysis results are not necessarily smooth. One immediately notices that none of the packages has consistent results with regard to CPU time and optimal design. Also, several packages failed to converge from one or more starting points. (The convergence failures are marked by an asterisk in the table.) NEWSUMT-A, though, has the most consistent results with regard to optima with CPU times comparable to those of DOT-SQP. However, DOT-SQP converges consistently on a better optimum than any other package, but is slower than DOT-MFD.

Table 2 Performance of Optimization Methods – RS model

	1	2	3	4	5	6	7	8	9	10
	Feasible starting points					Infeasible starting points				
Method										
DOT—MFD	187	300	239	209	252	165	169	130	137	125
	281	475	380	320	425	211	262	191	173	190
	622936	622873	622802	622803	622810	622817	622859	622841	622957	622808
DOT—SQP	50	43	47	41	47	78	81	44	46	37
	51	59	59	44	67	64	38	47	60	47
	622837	622825	622840	622854	622839	622836	622859	622824	622830	622834
FFSQP	2734	5872	5019	5021	5035	4185	6375	4439	4122	5124
	22	42	42	42	42	36	51	37	36	42
	622937	622938	622938	622938	622938	622938	622938	622937	622938	622938
LANCELOT	469	207	407	396	505	479	603	1001	583	313
	280	197	193	207	199	252	309	383	296	215
	622520	655699	626031	622502	622507	622510	622740	622548	622503	622423
NEWSUMT-A	492	531	523	534	470	515	518	524	537	552
	563	619	553	582	603	603	604	611	584	612
	622829	622831	622830	622830	622831	622829	622830	622830	622830	622831

FFSQP is still the slowest with CPU times between 13.6 and 35.4 hours. LANCELOT no longer converges to the best optima, and its CPU time, on average, is still longer than that for DOT.

A comparison of Tables 2 and 3 shows clearly that *none of the optimization packages can be inserted directly into the MDO design loop.* Numerical noise (due to discontinuous paneling schemes, adaptive algorithms, roundoff error, etc.) in the function values returned by the HSCT analysis codes results in both slow and erratic convergence behavior of all these derivative-based codes. The response surface methodology has two salient features: (1) *numerical noise is effectively filtered out*, resulting in faster and more consistent convergence behavior of the optimization codes, and (2) *the cost of expensive analyses is significantly reduced*, since only a few carefully selected expensive analyses are conducted to generate the response surface approximations. Furthermore, the response surface approach lends itself well to the use of multilevel, variable-complexity modeling as in Burgee et al[2].

CONCLUSIONS

In summary, all the packages tested are easy to use and relatively easy to implement. DOT and LANCELOT are very well documented. For the response surface model in the HSCT code the DOT-SQP optimization technique is the fastest by a wide margin and FFSQP is the slowest. LANCELOT converges to the best optima but with constraint violations that account for the better optima. None of the derivative-based optimization packages can be incorporated directly into the MDO design loop, because none of them deals effectively with noisy function values. A response surface methodology is effective both for improving the

Table 3 Performance of Optimization Methods – detailed model

	1	2	3	4	5	6	7	8	9	10
	Feasible starting points					Infeasible starting points				
Method										
DOT—MFD	916	716	1069	335	547	851	751	651	336	564
	226	150	272	61*	111	172	198	163	73*	62*
	638750	654636	639260	650000	655264	646954	630405	633700	633182	632572
DOT—SQP	2633	1599	563	2454	1678	1255	88	2070	1601	964
	247	29	94	237	232	180	6*	277	70	100
	636259	664025	631631	627487	628699	629875	650706	628273	626970	632595
FFSQP	34.9h	18.0h	28.4h	34.4h	35.4h	17.9h	20.3h	25.4h	35.6h	13.0h
	122	64*	100*	127*	134*	65*	51*	68*	133*	50*
	627008	652125	633599	635198	631212	642402	627057	635942	637181	629004
LANCELOT	4254	2300	4022	2111	1580	2401	2895	10515	4509	3138
	328	331*	276*	217	246*	304*	291	807	553*	341
	634472	650802	647559	658366	648880	654264	666670	644441	720769	642980
NEWSUMT-A	2361	1885	3218	1765	1865	1265	1731	2519	3603	1419
	761	822	549	594	614	331	417	841	839	368
	632814	627882	644212	642136	637605	644613	645183	629123	638255	638623

performance of optimization codes and for reducing the number of expensive analyses required for optimal design.

ACKNOWLEDGMENTS

We wish to thank all the authors of the optimization packages, and Nick Gould in particular, for making their codes available and for patiently answering questions about their codes. This work was supported in part by NASA Grant NAG-1-1562.

REFERENCES

1 Hutchison, M. G., Unger, E. R., Mason, W. H., Grossman, B. and Haftka, R. T., Aerodynamic Optimization of an HSCT Wing Using Variable-Complexity Modeling, *AIAA Paper,* **93-0101** (1993)

2 Burgee, S., Giunta, A. A., Balabanov, V., Grossman, B., Mason, W. H., Narducci, R., Haftka, R. T., and Watson, L. T., A Coarse Grained Parallel Variable-Complexity Multidisciplinary Optimization Paradigm, to appear in *Intl. J. Supercomputing Applications and High Performance Computing,* (1996)

3 Box, M. J. and Draper, N. R., Factorial Designs, the $|\mathbf{X}^T\mathbf{X}|$ Criterion, and Some Related Matters, *Technometrics,* **13,** 731-742 (1971)

4 Vanderplaats, G. N., *DOT User's Manual, v4.2,* Vanderplaats Research & Development, Inc., Colorado Springs, CO 80906 (1995)

5 Zhou, J. L. and Tits, A. L., User's Guide for FFSQP Version 3.5, Electrical Engineering Department and Institute for Systems Research, University of Maryland (1995)

6 Conn, A. R., Gould, N. I. M. and Toint, P. L., LANCELOT: A Fortran Package for Large Scale Nonlinear Optimization, Springer-Verlag (1992)

7 Grandhi, R. V., Thareja, R., and Haftka, R. T., NEWSUMT-A: A General Purpose Program for Constrained Optimization Using Constraint Approximation, *ASME J. Mechanisms, Transmissions and Automation in Design,* **107**, 94-99 (1985)

8 Fiasco, A. V. and McCormick, G. P., Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley and Sons, Inc. (1968)