SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

No. 7 Fall 1995

THE OPTIMIZATION TECHNOLOGY CENTER

by Steve Wright

All of us would like to see optimization used more widely and more effectively by applications scientists, engineers, economists. We have all fielded questions along the lines of "Here's my problem — I think it's an optimization problem — can you help me solve it?" Sometimes these questions have led to interesting collaborations, new research directions in optimization, even to new careers. Sometimes, however, we consign such questions to the "too hard" basket because they don't quite fit any of the familiar paradigms and, anyway, we don't have time. Often, the question isn't even asked in the first place. Too many scientists and engineers are unaware that optimization could help them solve their research problems faster and better than the methods they are using now.

In making the transition from theory and algorithms to applications, linear programming (discrete and continuous) has been the outstanding success story of optimization. Why have other areas, such as optimization of nonlinear or stochastic models, been slow to follow suit? Some of the reasons are inherent to the nature of the applications and algorithms, problems that can only be solved by patient research and cooperation. But there are other barriers to the use of optimization that are merely practical and logistical, things that we can do something about right now. These include

- It's too hard to find up-to-date information about algorithms and software;
- It costs too much time and money to get hold of the right software, install it, and make sure your computational environment can handle it;
- There are misunderstandings about what optimization software can actually achieve ("Why don't you find the global minimum? Why does my function have to be smooth?");
- It's sometimes difficult to interface to optimization software (for example, to represent the application in FORTRAN, to write derivative evaluation code, etc.).

The recent explosion of interest in the Internet gives us a great opportunity to jump some of these barriers. Millions of people now turn to the World Wide Web as their primary source for information on unfamiliar subjects. By making optimization information and services available through this medium, we can greatly expand our user base.

The Optimization Technology Center was founded in October, 1994 to grasp this opportunity. We aim to use the Internet and other developments in computing and communications to make optimization technology more accessible and easier to use.

OTC membership includes a small group of researchers at Argonne National Lab and Northwestern University, together with affiliated postdocs, graduate students, and undergraduates. Most of the OTC's efforts are devoted to development of the Network-Enabled Optimization System (NEOS), an Internet-resident source of optimization information and services. NEOS currently consists of two components: the NEOS Server and NEOS Guide.

The NEOS Server is a facility for solving optimization problems remotely over the Internet. Look for it on the Web at `http://www.mcs.anl.gov/home/otc/Server/`. Users send problems to the Server via email, ftp, or the Web; results are returned similarly. The Server unpacks and interprets the user input, schedules the job on one of our workstations (or on a node of Argonne's IBM SP multiprocessor), and calls a mini-Server to do the processing. There is one mini-Server for each problem area.

The Server aims to make it as easy as possible for the user to set up their input. For instance, the unconstrained optimization model requires only two FORTRAN routines, one to evaluate the function and one to define a starting point. Users can also supply a gradient evaluation routine if they choose, but it's often easier to let the Server find its own derivatives with help of the automatic differentiation tool ADIFOR.

Besides unconstrained minimization, the Server currently handles linear programming, stochastic linear programming, bound-constrained minimization, and network linear programming. More than one code is available for most problem classes. Both codes written by OTC members and existing codes (such as NETFLO and RELAX-IV for network programming) are supported.

Contents:

Server development is proceeding apace. We are developing a tool that allows users to connect directly to the Server through a unix socket from their own workstation, and a branch of the Server that interacts with the AMPL modeling system. Another extension will see the Server acting as an intermediary between optimization code developers on the one hand and users on the other. The Server will pass data from users to the code developer's workstation and activate a process on that machine, then pass the results back to the user.

The NEOS Guide can be found on the Web at `http://www.mcs.anl.gov/home/otc/Guide/`. It is still very much under development (many things remain to be filled in) but it already contains a lot of useful information for beginners and experts alike. There is an online version of the *Optimization Software Guide* (Moré and Wright, SIAM, 1993), updated where necessary to include new codes. There is the "Optimization Tree," a collection of short descriptions of various topics in optimization, with pointers to relevant software and sources of further information. For the benefit of software developers, we are building up a comprehensive test problem collection, which includes, where possible, documentation on the origin of each problem and its underlying application. The Guide also includes a set of case studies, one for each major area, which guide the uninitiated through the process of examining a practical application, formulating it as an optimization problem, solving it with optimization software, and interpreting the results. Information in the Guide can be accessed easily through various search and navigation tools.

Looking beyond the Server and Guide, we plan to establish the OTC as a place where high-quality software tools are produced as well as disseminated. These tools will be collected in the NEOS Library, which will emphasize such design features such as component reusability and modeling language interfaces. We also seek alliances with key people in applications areas so that, through them, we can demonstrate the usefulness of optimization technology to entire communities of researchers.

Obviously, much of the hard work in the construction of NEOS lies outside the scope of a traditional academic research program. We have had to learn some new skills, including Web page design, graphic design, and systems programming in the scripting language Perl. Since we are a small group of busy people, all this takes time. And there is the question of talent — my own taste in graphics has been called into question on one or two occasions. Nothing would have been possible without our dedicated undergraduate interns, who have done much of the hard work of programming the server and beating the Web pages into shape.

We welcome collaboration in all aspects of the OTC's activities. If you are interested in contributing information to the Guide or codes to the Server, get in touch! Our operators are standing by to take your call.

CHAIRMAN'S COLUMN

by Jorge J. Moré

All the nominations for the SIAG/OPT prize are in. Choosing a winner will be difficult, but the selection committee is outstanding, and represent a wide range of interests. The members of the selection committee are Tim Kelley (chair), Clyde Monma, Mike Powell, Bobby Schnabel, and Mike Todd. The winner is scheduled to be announced at the SIAM Optimization meeting, which will be held May 20–22, 1996, in Victoria, British Columbia.

If you want more information on the SIAM Optimization conference, look at the Web page for the conference:
`http://`
`www.siam.org/meetings/op96/op96home.htm`
You will almost certainly like this page. I hope that you can make it to the conference.

As part of our effort to advertise the SIAG/OPT prize, SIAM sent out an email message to each member of SIAG/OPT requesting that they consider nominating a paper for the prize. I was not aware until recently that we had this ability. We will use this facility to keep you informed of occasional items of interest to the membership, but for the moment we do not have plans to expand in this direction.

You may remember that I proposed setting up a Web page on interesting optimization problems. Now, thanks to SIAM, we are developing a Web page for the SIAG/OPT. The initial design was done by the SIAG/OPT officers, with the cooperation of Laura Helfrich at SIAM. In other words, we made suggestions, and she did all the work.

We are seeking suggestions for what to include in the SIAG/OPT Web page. At present we have the following headings on the main page:

- Conferences
- Prizes
- Optimization Sites
- Software
- Newsletters, FAQ's, and bibliographies
- Journals
- Books

We already have pointers in each category, but we need suggestions for additional pointers. You can send suggestions to any of the officers. In case you have mislaid this information, these are the addresses:

Jorge Moré more@mcs.anl.gov

Tim Kelley, tim_kelley@ncsu.edu

Ekkehard Sachs, sachs@uni-trier.de

Philip Gill, pgill@ucsd.edu

The main listing under Conferences is the SIAM Optimization conference. We would like to list other conferences of potential interest to SIAG/OPT members under this heading. Suggestions?

Under Optimization Sites we have pointers to Web pages with a strong optimization content. I realize that this can lead to a large number of pointers, but at present we are not worried about this. We need suggestions.

The Software heading has pointers to Web sites that feature optimization software. These are generally repositories with large amounts of software, but we are also interested in Web pages that are strongly related to optimization software.

The Newsletters, FAQ's, and bibliographies entry is self-explanatory. We only have a couple of bibliographies, and I suspect that there are more.

The Journals entry is interesting. We have pointers for Web pages with table of contents for journals of general interest to optimizers. At first I thought that there would be a large number of entries under this heading, but interestingly enough, there are just a few pointers that are not SIAM journals. We could only find the ACM Transactions on Mathematical Software, and the IEEE Computational Science and Engineering journal. I regularly see table of contents published in electronic newsletters, but they do not list Web addresses for the table of contents. Apparently SIAM is way ahead of other publishers in the Web business. By the way, the suggestion to have table of contents came from Henry Wolkowicz.

What other entries would you like to see? Suggestions are welcome.

I would also like to have a section on Optimization Applications. This would contain pointers to uses of optimization in applications. We would like to have informative displays that show how optimization is used in applications. We also would like to have some eye-popping pictures associated with these pages. We should not emphasize glitter over substance, but in the Web, glitter is important.

I encourage you to point your viewers to the SIAG/OPT Web page at
http://www.siam.org/siags/siagop/siagop.htm
and provide suggestions for additions (or deletions). If this project is to succeed, suggestions will have to come from the SIAG/OPT members. The SIAG/OPT page should reflect the interests of the membership.

Unfortunately, I have to end this column by announcing that Larry Nazareth has decided to step down as editor of the SIAG/OPT Newsletter: Views and News. He feels that new blood must be injected into the Newsletter, and that he must make room for a new editor. As I said in my previous column: Larry Nazareth has been doing an outstanding job as editor, and his efforts have led to a classy publication. His contributions will be missed. Replacing him is difficult. We are still searching, and hope that by the time this newsletter reaches you, we will have found a new editor.

FORUM ESSAYS

INVITED AUTHOR'S PREVIEW: NONLINEAR PROGRAMMING

by Dimitri P. Bertsekas [1]

Nonlinear programming is a mature field that has experienced major developments in the last ten years. When I reflect on the evolution of the subject over the last 30 years, I find that by the late sixties and the early seventies most of the major ideas were already available in some form. I am thinking here of the topics that still collectively form the foundations of the field: iterative descent methods and their convergence analyses, conjugate directions, Newton/Quasi-Newton methodology, stochastic gradient methods, penalty, interior point and augmented Lagrangian methods, duality and convex programming, nondifferentiable optimization and large-scale problem decomposition. Moreover, the understanding of these topics matured in the subsequent decade. Nonlinear programming became a mainstream subject that is covered in most schools' graduate curricula, general purpose nonlinear programming codes became available, and nonlinear optimization was accepted as the primary methodological vehicle in an ever broadening spectrum of important practical applications.

As the mid-eighties were approaching a stage was reached where many researchers felt that nonlinear programming methodology had matured and had reached a steady-state. Yet there have since been several developments that have resulted in substantially new perspectives. The first such development is the merging of linear and nonlinear programming algorithms through the use

[1] Lab. for Information and Decision Systems, Massachusetts Institute of Technology, Room 35-210, Cambridge, MA 02139. email: dimitrib@mit.edu

of interior point methods. This has resulted in a profound rethinking of how we solve linear programming problems, and in a major reassessment of how we treat constraints in nonlinear programming. A second development, less visible but still important, is the increased emphasis on large-scale problems, and the associated algorithms that take advantage of problem structure (such as the presence of a network or a dynamically controlled system) as well as parallel hardware. A third development has been the emergence of neural networks as a very popular technology in a broad variety of practical contexts. The training of neural networks and other related approximation architectures often gives rise to challenging singular least squares problems that require trial and error, and expertise in nonlinear programming. Somewhat paradoxically, these problems are better dealt with simple (possibly stochastic) gradient-like methods and stepsize rules, rather than sophisticated Newton-like methods, thus significantly changing our perspective of iterative descent algorithms.

While teaching nonlinear programming over the last twenty years, I have been developing a set of class notes that has finally become a book[2]. I am happy that the major developments of the last ten years have been reflected in the book, although I am sure that much further progress lies ahead.

The purpose of the book has been to provide a fairly comprehensive and mathematically rigorous account of nonlinear programming at the beginning graduate student level. Classical topics, such as descent algorithms, Lagrange multiplier theory, and duality, are covered. In addition, interior point methods for both linear and nonlinear programs are treated in detail, the major aspects of large-scale optimization are developed, and least squares problems are discussed extensively, including their solution by simple incremental gradient-like methods that are commonly used in neural network training problems.

Unconstrained optimization is covered extensively in a beginning chapter: optimality conditions, gradient and Newton-like algorithms, conjugate directions, nonderivative methods. The material is classic, but there are discussions of topics frequently left untreated, such as the behavior of algorithms for singular problems, neural network training, and discrete-time optimal control.

I have chosen to first discuss constrained optimization over a convex set without introducing the complicated Lagrange multiplier machinery. I have found that students like this approach, which views algorithms such as conditional gradient, gradient projection, the affine scaling method for linear programming, and coordinate descent, as natural extensions of unconstrained descent algorithms.

A nonlinear programming instructor has to choose the level and the framework for treating Lagrange multipliers and duality. One may use a variational approach based on the implicit function theorem, or a convex analysis approach based on convex analysis/support hyperplanes. The former approach applies to a broader class of problems, while the latter is more elegant and more powerful for the convex programs to which it applies. I have covered both lines of analysis, but I have also provided the option for a third approach that I have on occasion followed in class, because it is more economical in terms of lecturing time. In particular, if one develops Lagrange multiplier theory for linear constraints (using Farkas' lemma) one may develop in a simple way a fairly powerful form of duality theory for linearly constrained problems with differentiable convex cost, including linear and quadratic programming.

The algorithmic treatment of constrained problems includes barrier, augmented Lagrangian, sequential quadratic programming, and primal-dual interior point methods for linear programming. There is also an extensive treatment of nondifferentiable optimization, including subgradient, ϵ-subgradient, and cutting plane methods. Decomposition methods such as Dantzig-Wolfe and Benders are also discussed.

In most scientific books the author has to make coverage compromises. The subject of nonlinear optimization has grown so much that leaving out a number of important topics was inevitable. For example I had to forego a discussion of variational inequalities and a deeper treatment of Quasi-Newton methods.

Similar to the early eighties, many researchers today feel that nonlinear programming has reached a steady-state. As an author I hope so, although as a researcher who has experienced the depth and rich variety of the subject and has witnessed its evolution for many years, I strongly doubt that this will prove to be the case.

****

[2] "Nonlinear Programming," 640 pages, published by Athena Scientific, Dec. 1995 – information and table of contents may be obtained by email from the publisher (athenasc@world.std.com) or the author (dimitrib@mit.edu), and also from the author's www page http://web.mit.edu/dimitrib/www/home.html

UPROOTING SIMULATED ANNEALING AND GENETIC ALGORITHMS

by Bennett L. Fox [1]

1. INTRODUCTION

The feature article, *New optimization methods rooted in biology and physics*, in the Spring 1995 issue of *Views-and-News* outlines the respective biological and physical origins of two global optimization methods: *genetic algorithms* (GAs) and *simulated annealing* (SA). They have been handicapped by slavish mimicking of those respective origins. While those roots have historically served as points of departure, the time has come to cut them off — as we sketch in Section 2. Thus, we decouple simulated annealing and genetic algorithms from their respective physical and biological origins. Section 3 discusses noisy observations (even when the underlying problem is deterministic), traditionally not considered in the setting of GAs and SA.

The proper backdrop for *all* probabilistic search methods, including GAs and SA, is Markov chains — on a sufficiently *rich* state space. That setting subsumes the most important feature — multiple solutions in a state (*population* in genetic-algorithm jargon) — of GAs, while — unlike them — having no duplicate copies of feasible solutions in a state. We use an entirely different move mechanism; see Fox [3,6] for details.

Candidates for the next state can be generated from respective individual solutions in the current state by *local search*, by combining solutions in the current state (*crossover* being but one such way), and by randomly generating solutions from the entire feasible domain (*mutation* being but one such way). Fox [3,6] details a scheme for doing this for finite and general domains, respectively, while integrating the transition mechanism with both SA and the key feature of *tabu search* (TS): incorporating short-term memory into each state and a corresponding penalty function. A major reason for the latter is to inhibit short-run oscillation when *proposed* moves take account of objective-function values in the current neighborhood. The TS literature abounds with ingenious penalty functions reflecting additional criteria. As far as we know, TS has no counterpart in nature — perhaps explaining its absence from the feature article cited above. See Glover and Laguna [13] for an authoritative account of tabu search. The volume in which reference [3] appears is devoted to tabu search. Glover [12] *linearly* combines solutions, which neither standard GAs nor nature do, followed by what he calls *directional rounding*. Together with certain additional ideas, he calls the result *scatter search* and indicates that it can be combined with other heuristics. He notes that "relying too literally on the genetic metaphor" excludes scatter search (and, we add, many other useful ideas). Links among scatter search, GAs, TS, structured combinations, and relaxation are explored in Glover [10,11]. Fox [3] compares mutation with neighborhood enrichment by generating (a starting point for) a proposed move *uniformly* over the *entire* state space, demonstrating an advantage of the latter (despite the increased short-run computation it usually requires).

Neither the GA nor TS community sees search through a Markov-chain lens. Lacking this *weltanschauung* makes it hard to formulate certain theorems, let alone prove them. In sharp contrast, SA generally is formulated explicitly as a Markov chain — though typically on an unduly restrictive state space (which SA theory does *not* require).

2. CUTTING OFF ROOTS

Naive SA. A straightforward SA approach (rigidly) mimics physical annealing in several ways:

1. *Bare-bones state space.* States are in one-to-one correspondence with feasible solutions.

2. *Blind moves.* Moves are proposed with little or (more commonly) no regard for objective-function values.

3. *Explicit rejection.* Proposed moves are tested for acceptance and, if rejected, the state is unchanged.

4. *Myopic search.* Proposed moves are restricted to a "local" neighborhood.

5. *No look-ahead.* No descent (usually but not always [Torczon [22], for example] gradient-based when the domain when the domain is continuous) from a proposed starting point.

6. *No preprocessing.* The initial state is arbitrary.

None of these characteristics of physical annealing are necessary for simulated annealing, as Fox [3,6] details. He argues that none are desirable. When strategies for global optimization are compared, often *naive* SA is used as a straw man.

Smart SA. Partly following Fox [3,6], we briefly comment on the respective points above:

1. *Fleshed-out state space.* A richer state space is needed to incorporate key features of GAs and TS in (uprooted) SA.

[1] Mathematics Department, Campus Box 170, University of Colorado, P. O. Box 173364, Denver, CO 80217-3364, email: bfox@castle.cudenver.edu. Thanks to Larry Nazareth for inviting this contribution.

2. *Smart moves.* We take off the blinders. Without dependence on objective-function values, there is no way to discriminate among improving moves. When the domain is continuous, some have proposed certain weak dependence on objective-function values but nevertheless generate the coordinates of a proposed point *independently*; even if axes are rotated in some problem-dependent way, it is easy to see *a priori* that in general generating coordinates *independently* is ineffective (because some valleys are not "aligned" with the axes).

3. *Loop skipping.* An alternative move mechanism advances simulated time to the epoch of the last move L in respective self-loop sequences of rejected moves (of the form $x \rightarrow x \rightarrow \cdots \rightarrow x$) and then, given L, routinely generates the state at epoch $L+1$. Unlike previous approaches, Fox [3,5,6] does this without obliterating the given cooling schedule — while doing it in $O(1)$ *computer* time. Fox [5] and the pair of papers, Fox and Heine [8] and Heine [14], respectively, detail the quadratic-mean and almost-sure senses in which this assertion holds. Heine [15] examines the Cesàro-sense sample-path implications of advancing simulated time by simply generating a geometric variate with parameter depending on the current state and temperature. With the neighborhood structure recommended in Fox [3,6] but (as far as we know now) not in general, the probability that the current state is optimal converges to one using the same simple method of advancing time. Unlike the straightforward implementation of the SA acceptance test, Fox's scheme and *a fortiori* the simple method are compatible with parallel processing on SIMD computers. Loop-skipping, as above, is a shortcut that nature cannot take.

4. *Dynamic enrichment.* Restriction to a "local" neighborhood makes the search unduly myopic. Putting a few time-dependent non-local states in the current neighborhood *diversifies* the search. In other words, it increases search *breadth.* Fox [5,6] shows, via a non-pathological example, that purely local neighborhoods can asymptotically disconnect the state space and exhibits a bad consequence. For theoretical purposes *only*, he recasts time-dependent neighborhoods as static neighborhoods on a more elaborate state space. On that space, from each state there is a path *with no uphill moves* to the set S_0 of optimal states. When a path depends on the non-local neighborhood enrichment indicated above, generally it has low probability — except possibly at very low temperatures. Making these dynamic neighborhoods *small* makes proposing moves influenced by objective-function values practical; likewise, for the alternative move mechanism above and look-ahead below. *Candidates* for these neighborhoods come (generally not uniformly) from the entire space, as Fox [3,6] details. Not all candidates succeed in entering the neighborhood.

5. *Look-ahead.* When one tries to propose moves *intelligently* (implying discrimination among improving moves, for example), look-ahead is natural. Though the starting point for a proposed move comes from an enriched neighborhood, descent from it is relative to the local (unenriched) neighborhood. For discrete domains, look-ahead is just a heuristic. In contrast, when it is continuous, it is easy to see that — without look-ahead — ordinarily the convergence rate is glacial (because it becomes increasingly hard to generate a proposed point better than the current best point — especially when the (local) valleys have steep walls but small volumes near their respective bottoms). For readers of *Views-and-News*, an added attraction of using a descent subroutine is that it links SA to mainstream mathematical programming. The more intense the look-ahead, the greater the search *depth.* Trading off search depth and search breadth, along with specifying neighborhoods (and — in our setup — tabu penalties), is the *art* of SA. Intelligent neighborhood specification, when possible, makes it likely that "deep energy wells also drain wide basins" — a necessary (and nearly sufficient) condition for SA to work well, as noted in the feature article in *Views-and-News* cited above, quoting Kauffman [16], p. 112. A caveat is that, with the way we dynamically enrich the neighborhoods, the added drains may be narrow (hence hard to find) if the landscape with local neighborhoods does not satisfy Kauffman's condition. Though with dynamic enrichment there is only one well and it drains everything, no approach is a panacea. Nature does not have freedom to choose its neighborhoods, but SA does.

6. *Preprocessing.* While foreign to nature, preprocessing is a mathematical ally in eclectic algorithms. Combined with the spatial memory induced by having multiple solutions in a state, it gives the algorithm a feel for the landscape by recording scanned points at or near deep valley bottoms. Only after getting that global feel, the algorithm explores intensively promising regions with possible uphill moves. More than its name perhaps suggests, preprocessing is a prominent part of our meta-strategy and, depending on the termination criteria, may dominate the overall computation — with simulated annealing proper becoming a postprocessor. It can be random restarting in tandem with descent. This as a stand-alone technique beats *naive* SA in the

sense that, for all large enough k, the probability that an optimal state is scanned by *simulated* time k is at least large under the former. However, using Fox's loop-skipping algorithm and (thus reasonably) counting only accepted moves, Fox [4] shows by example that is no longer true in general. Stratifying the restarts makes sense. In a continuous-domain counterpart, where the state space (or a projection of it) is the unit cube, Fox [6] shows how to team descent with so-called *(t, m, s)-nets* in a natural way, producing *clusters* analogous to those used in certain other approaches to global optimization but which — unlike those — are *not ad hoc*. The (excellent) standard reference for such nets (apparently little known in the optimization community) is Niederreiter [19]. In this setting (as in fixed-dimensional numerical integration), these nets are much better spaced (for example, relative to the largest "hole") for a given number of points than grids or a set of points intended to mimic truly (independent) uniform random numbers. Fox [6] deliberately chooses the coordinates of the point to which a move is next proposed far from independently by first choosing a "box" (in the net) to scan next via a dynamically-updated weight vector. That box contains the point next scanned. It is natural to start with a coarse net $\mathcal{N}$ and then, adaptively, refine it by laying down nets on certain boxes of $\mathcal{N}$. This is roughly analogous to adaptive numerical integration and to adaptive multilevel techniques for PDEs (Rüde [21] for example). Another weapon in the preprocessing arsenal gets lower bounds on the objective function over various respective regions, eliminates those for which the bound exceeds a known feasible solution, and points to those for which the lower bound falls much below any known feasible solution as especially attractive to explore.

Recap. It is unproductive to get wedded to just one of SA, GAs, and TS. Far from being disjoint, they can be integrated into a coherent whole (as indicated above and, in more detail, in Fox [3,6]), which is much more than the sum of its parts. That whole stands on its own, decoupled from nature. It contains a powerful preprocessor, unlike the usual implementations of SA, GAs, and TS separately. We view that hybrid as *smart* SA, but a good case can be made to view it as a meta-strategy *sui generis*.

Implementation. Kawai [17] is implementing our ideas for unit-cube domains. The user has to supply only a subroutine that computes the objective function and, depending on the local-search subroutine, certain of its derivatives. With the default conjugate-gradient subroutine, the user-supplied subroutine must (of course) compute two derivatives. In addition, the user has the option of overriding default values of certain tuning parameters. Though Kawai's (experimental) highly-modularized code is limited to low dimensions, it scales easily to (much) higher dimensions — except that then the (t, m, s)-nets would have to generated differently in a practical implementation. Currently, they are extracted from Niederreiter's (1988) base-2 (t, s)-sequence. Because of the corresponding fast growth of $t(s)$ in the dimension s, in higher dimensions — if extraction from (t, s)-sequences is to be used — different bases $b(s)$ giving a slower growth of $t(s)$ are available from Bratley, Fox, and Niederreiter [1],[2] (though not recommended in those references for numerical integration). Alternatively (and in principle better), (t, m, s)-nets can be generated directly (Mullen, Mahalanabis, and Niederreiter [18] for example). Perhaps best of all, (t, m, s)-nets can be extracted from the new base-2 Niederreiter-Xing [20] (t, s)-sequence, for which $t(\cdot)$ grows much more slowly than with any other known base-2 (t, s)-sequence. Niederreiter and Xing are currently working out details that would make implementation practical. Note that $m > t$, the number of points in the net is b^m, and that — for fixed b — the smaller t, the better the spacing. For discrete domains, there is currently no implementation — partly because (though certain subroutines can be written in a problem-independent way) significantly more *problem-dependent* code is needed. The increased tailoring required for discrete problems appears in all approaches.

Speed-up on parallel computers. Nature must work with just one copy of a physical system. In contrast, mathematical counterparts (such as SA) to nature's optimization schemes can work with many. This can produce dramatic speed-up on parallel computers of the expected time to first hit the set S_0 of optimal states or any non-empty set of the form $\{s : c(s) \leq b\}$, where c is the objective function. Fox [5] characterizes this speed-up mathematically and, applying linear algebra and some analysis to the Markov-chain setup (also relevant to certain variants of TS), gives sufficient conditions for *linear* speed-up in a precise sense, when the domain is finite. Extending this result to (certain) general domains is an outstanding open problem.

GAs. In working with biological systems, especially in the setting of *evolution*, nature does work with multiple copies. Yet nature's progress is slow. We think that the explanation is that nature's move mechanism is awkward. This is reflected in our reinterpretation of the following folklore: the more faithfully GAs try to imitate nature (variants without local search, for example), the less successful they are empirically. The key ideas of GAs can be uprooted and incorporated in SA, as we have indicated above, with a flexible move mechanism uninhibited by nature's constraints.

Neural networks. Neural networks (NNs), also mentioned in the feature article cited above, are not part of the package. There are two reasons for this. First, the mathematics of NNs seems orthogonal to it. Second, NNs seem so tightly bound to their biological and physical antecedents that, to be (possibly) competitive as an optimization technique, vastly different computer architectures (perhaps inspired by human brains) — not yet available — are needed.

3. NOISE: INTRINSIC AND INDUCED

The objective-function c may be measured with noise or it may be computed inexactly, with the error in the estimate $\hat{c}(s)$ going to zero almost surely as the number of visits to state s increases. For example, in either case, $c(s)$ may be the expectation some random variable $Z(s)$ which we can simulate. In the second case, this may be a profitable viewpoint if $c(s)$ is the sum of (perhaps combinatorially-explosively) many terms. *Importance sampling* may well be effective here. More generally, we can simulate far more efficiently than nature does; see Fox and L'Ecuyer [9] for example, where — as a special case — $c(s)$ can be the reciprocal of simulation "efficiency" using simulation strategy s.

Denote the state space by S, for practical reasons generally orders of magnitude smaller than with noiseless observations. Our adapted SA procedure is the natural one:

> Base the current move on the current estimates $\{\hat{c}(s) : \forall s \in S\}$.

This is a situation where, while it is helpful to reduce the expected time to hit first hit S_0, the real goal is to hit S_0 frequently — to refine our estimate of c on S_0 as quickly as possible. Fox and Heine [8] show, under arguably-reasonable hypotheses, that the procedure above inherits the convergence-in-probability of the (hypothetical) counterpart where all $c(s)$ are observed without noise. When S is finite, Fox and Heine [7] give compatible estimators. Extension to more general S, via likelihood ratios, is examined in Fox and L'Ecuyer [9] . For general S, *stochastic approximation* plays a role in local search. Extending a result in Fox [6] for finite S, Fox and L'Ecuyer [9] find — for general S — an asymptotically-valid confidence interval for $\min\{c(s) : s \in S\}$ which — under certain conditions — is as good as if $\arg\min\{c(s) : s \in S\}$ were known in advance. The practical point is that *crude* estimates of those $c(s)$ which are *not* top contenders suffice and that, for large but finite computer-time budgets, this — along with good estimates for the top contenders — is what SA produces. The *bandit* approach, popular among statisticians, does likewise, but — unlike SA — does not exploit structure on S. Such structure can be induced by computationally-exploitable connections (perhaps reflecting a suitable metric) or by *a priori* considerations suggesting (but not proving) a certain partial order on S relative to c.

References

[1] P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, 2:195-213, 1992.

[2] P. Bratley, B. L. Fox, and H. Niederreiter. Algorithm 738: Programs to generate Niederreiter's low-discrepancy sequences. *ACM Transactions on Mathematical Software*, 20:494-495, 1994.

[3] B. L. Fox. Integrating and accelerating tabu search, simulated annealing, and genetic algorithms. *Annals of Operations Research*, 41:47-67, 1993.

[4] B. L. Fox. Random restarting versus simulated annealing. *Computers and Mathematics with Applications*, 27:33-35, 1994.

[5] B. L. Fox. Faster Simulated Annealing. *SIAM Journal on Optimization*, 5:488-505, 1995.

[6] B. L. Fox. Simulated annealing: folklore, facts, and directions. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P. J.-S. Shiue, eds.), Lecture Notes in Statistics #106, Springer-Verlag, Berlin, 1995, 17-48.

[7] B. L. Fox and G. W. Heine. Stable estimators for self-adjusting simulations. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P. J.-S. Shiue, eds.), Lecture Notes in Statistics #106, Springer-Verlag, Berlin, 1995, 205-212.

[8] B. L. Fox and G. W. Heine. Probabilistic search with overrides. *Annals of Applied Probability* (to appear).

[9] B. L. Fox and P. L'Ecuyer. Efficiency improvment for Monte Carlo simulation. In preparation.

[10] F. Glover. Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics*, 49:231-255, 1994.

[11] F. Glover. Genetic algorithms and scatter search: unsuspected potentials. *Statistics and Computing*, 4:131-140, 1994.

[12] F. Glover. Scatter search and star-paths: beyond the genetic metaphor. *OR Spektrum*, 17: Issue 2/3, 1995 (to appear).

[13] F. Glover and M. Laguna. Tabu search. In *Modern Heuristic Techniques for Combinatorial Problems*(C. Reeves, ed.), Blackwell Scientific Publications, Oxford, 1993, 70-150.

[14] G. W. Heine. Convergence properties of a loop-skipping algorithm for simulated annealing. In preparation.

[15] G. W. Heine. Pathwise convergence for simulated annealing with an adaptive schedule. In preparation.

[16] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York, 1993.

[17] M. H. Kawai. Implementing sophisticated simulated annealing. MS project, Mathematics Department, University of Colorado, Denver. In preparation.

[18] G. L. Mullen, A. Mahalanabis, and H. Niederreiter. Tables of (t, m, s)-net and (t, s)-sequence parameters. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (H. Niederreiter and P. J.-S. Shiue, eds.), Lecture Notes in Statistics #106, Springer-Verlag, Berlin, 1995 (to appear).

[19] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* CBMS-NSF Regional Conference Series in Applied Mathematics #63, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[20] H. Niederreiter and C. P. Xing. Low-discrepancy sequences and global function fields with many rational places. Technical report, Institute for Information Processing, Austrian Academy of Sciences, Sonnenfelsgasse 19, A-1010 Vienna.

[21] U. Rüde. *Mathematical Computational Techniques for Multilevel Adaptive Methods.* Society for Industrial and Applied Mathematics, Philadelphia, 1993.

[22] V. Torczon. Pattern search methods for nonlinear optimization. *SIAG/OPT: Views-and-News*, No. 6:7-11, Spring 1995.

****

GLOBAL MODELING FOR OPTIMIZATION

by Paul D. Frank [1]

1. MOTIVATION FOR GLOBAL MODELING

The problem addressed is that of obtaining globally optimal, or at least globally "good", solutions of problems where function evaluations may require several hours of supercomputer time. In addition, the optimization problem may not be differentiable. This situation arises frequently in industry, where function evaluations may require runs of analysis codes in fields such as computational fluid dynamics or structural dynamics. Such codes are often expensive-to-run, black boxes from the user's perspective. In many cases, the user knows little about the smoothness of the analysis results with respect to the optimization variables, and does not have a good starting guess for the optimization process.

In the above situation, there is little to be gained by blowing the entire computing budget to obtain a local optimizer. Instead, it is more desirable to use the limited available resources to obtain insight into the global trends of the optimization functions. This global insight can be used to determine good designs, "screen" out ineffective variables, or identify regions in design space worthy of further exploration.

To obtain global insight and globally good designs, the method considered here is modeling over the design space. The models are based on data obtained by running the expensive analysis code at an affordable number of points, well spread out in the design space. It is assumed throughout that the designer can define reasonable bounds on the design variables.

In the current context, the number of data points will often be only a small multiple of the number of variables. In this situation, interpolating models are not likely to exhibit "wild" behavior. For example, there may only be 100 data points available for a 20 variable problem. In this case, there is not enough data to determine even a quadratic model. However, depending on the data and data locations, the interpolating models can describe higher-degree behavior in some subspaces. In addition, fidelity to the data is a key concern. Thus, interpolating models, rather than approximating models, are discussed in this report.

Analysis and design optimization using interpolating models have been applied to several problems at the Boeing Company. These problems include design of low-vibration helicopter rotor blades, design for low radar cross section, and optimal shot peen forming of wing skins (the process of curving and work-hardening aluminium

[1] Boeing Information and Support Services, P.O. Box 3707, MS 7L-21, Seattle, WA 98124-2207.
e-mail: frank@espresso.rt.cs.boeing.com

wing skin sections by hitting them with lead shot at high speeds).

The remaining sections discuss selection of data points, formation and analysis of interpolating models, optimization using interpolating models, and areas for further research.

2. DETERMINING WHERE TO RUN THE EXPENSIVE ANALYSIS CODE

The main requirement for selecting the points at which to run the expensive analysis code is that they be well spread out in the design space.

For many years, statisticians have studied the problem of selecting data points for physical experiments. Recently, they have addressed the case of "computer experiments". The main difference in the case of computer experiments is that there is no need for extra runs to enable estimation of experimental error. One class of methods for determining data points, or *sites*, for computer experiments are the *optimal* methods. These methods optimize either quantities related to the conditioning of the matrices associated with the interpolation problem, or a statistical estimate of the integral of the model's error, or a geometric measure such as minimizing the maximum distance between any point in design space and the nearest data site. (See, e.g., [2,3,10] for discussions of site selection methods.) All the optimal methods tend to produce data sites that are well spread out in design space.

Unfortunately, optimal site selection methods require solution of a global optimization problem that has dimension equal to the number of data sites times the number of design parameters. For example, 30 design parameters and 150 data sites requires solution of a 4,500 variable global optimization problem. In addition, these optimization problems can have huge numbers of local minimizers. This is because every permutation of a local minimizer's design parameters yields another local minimizer. Thus, the optimal site selection methods are only practical for relatively small problems.

The above difficulties can be avoided by using good and readily computable, nonoptimal methods. One such class of methods are orthogonal array-based Latin hypercubes [1]. Roughly speaking, these methods produce "gridlike projections" onto certain subspaces.

The above discussion relates to selecting data points for an initial global model. Many research issues involve selecting the number and location of additional data points, after an initial model has been formed. These issues are discussed subsequently. However, it seems reasonable to this author that a minimum number of data sites for an initial interpolating model is a multiple, say three to ten, of the number of design space dimensions.

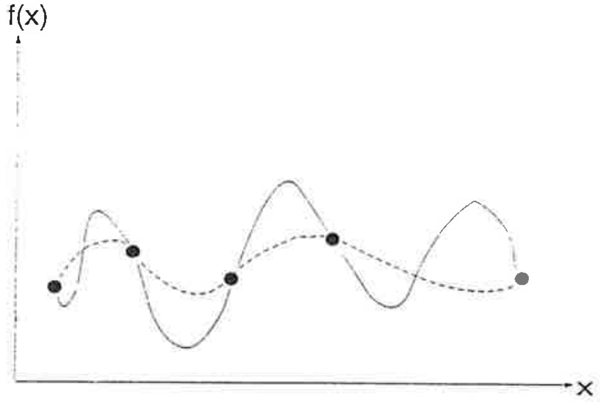
3. SELECTING AND COMPUTING AN INTERPOLATING MODEL

The criteria for selecting an interpolating model is that it be as robust and accurate as possible. In general, this means that the model be "taut", i.e. it doesn't introduce big oscillations that are not warranted by the data. For example, both the solid and dashed curves in Figure 1 interpolate the data points (indicated by large dots). However, it is clear that the dashed curve is the superior interpolator. One way to quantify this superiority is to say that the integral of the square of high-order derivatives is much smaller for the dashed curve than for the solid curve. In fact, there are modeling methods based on minimizing the "energy" in higher derivatives, such as the *thin plate spline* methods, see e.g. [6]. (They are called thin plate splines because they assume the minimum energy configuration of a material that has no resistance to deformation.)

Minimal *cross-validation* error is a property closely related to minimal energy in higher derivatives. At any data site, the one-at-a-time cross-validation error is the error at that point, for a model built using all data points except the given point. The total one-at-a-time cross-validation error is the sum of the one-at-a-time cross-validation errors at the data points. General cross-validation error is the total error obtained at deleted subsets of points, for all possible subset sizes.

A modeling method that produces a "wiggly" interpolator, such as the solid curve in Figure 1, would probably have a very large cross-validation error. This is because the large excursions are bound to yield high errors at some data points, when data at those points are left out of the interpolating model.

Figure 1



One interpolating model having the minimal cross-validation error is the maximum likelihood estimator (MLE) model from the statistical field of Design of Experiments (DOE) [3,2,4,8]. (Here, cross validation means the general, all possible deleted subset, type of cross validation.)

The DOE interpolating model $m(x)$ at a point $x \in \Re^n$ in design space has the form

$$m(x) = c(x)^T y, \qquad (1)$$

where $y \in \Re^m$ is the vector of dependent variable values at the data sites. Thus, at each at x, the value of the DOE model is a linear combination of the data values at the sites. The weighting factor vector $c(x)$ is determined from the statistical paradigm by selecting those correlation parameters that maximize the likelihood of having obtained the observed data values. A constraint on the maximization is that the model have the same mean as the data.

The intuitively appealing aspect of the DOE interpolating model is that the value of the model at any point is a weighted sum of the analysis values at the data sites. Also, if an exponential correlation function is selected, the weighting factor declines with distance from x to the data site. The rate of decline is determined by the data, via the MLE computation.

Forming the maximum likelihood DOE model requires global optimization over n correlation parameters. Each function evaluation for the optimization requires factoring an m by m, positive definite correlation matrix. However, once the DOE model is formed, evaluation of the model is quite inexpensive. The leading cost is computation of the correlations between x and the data sites.

Another robust method for forming interpolators is based on careful selection of polynomial basis functions [11]. The method only includes basis functions in the model if they are well-determined by data at the given site locations. Basis functions are successively added to the multidimensional polynomial model until all the data points are interpolated. The method favors low-degree terms over higher degree terms, but will choose higher-degree terms if the lower-degree terms are not well-determined. The above method will work for any number of runs and tends to produce low-degree interpolators.

4. ANALYSIS OF INTERPOLATING MODELS

One of the main uses of interpolating models is to gain insight into the global behavior of the underlying function. For example, the model can be used to estimate *main effects* and *interactions*. The main effect of variable x_i at a specific value $\bar{x}_i$, is obtained by setting $x_i = \bar{x}_i$, integrating the model over all the other variables, and subtracting the mean value of the model over the design space. Similarly, the interaction between variables x_i and x_j at specific values $(\bar{x}_i, \bar{x}_j)$ is obtained by integrating the model over all the other variables, subtracting the sum of the main effects of x_i at $\bar{x}_i$, x_j at $\bar{x}_j$, and subtracting the mean value of the model over the design space. Higher-order interactions are similarly defined. This is an illustration of an analysis of variance (ANOVA) decomposition (see e.g., [9].) In general, ANOVA decompositions can be used to determine the amount of the total design surface variation due to the main effect of each variable and each of the interactions among variables.

Plots of main effects and interactions can be used to eliminate ineffective variables from the problem, i.e. *screening*. They can also be used to locate interesting regions for further exploration. Use of the ANOVA decomposition is intuitively appealing for obtaining global trends of (possibly) nondifferentiable functions. This is because the ANOVA decomposition measures integrated effects and its existence requires only square integrability.

A statistical estimate of the model error at any point x can be computed for a DOE interpolating model (see e.g., [4,5]). Basically, the predicted model error increases with increasing distance from x to the nearest data sites, and increases with the nonlinearity exhibited by the data. Model error estimates provide one useful measure for determining where to take new data.

5. MODELING AND OPTIMIZATION

Since an interpolating model is comparatively cheap to evaluate, it is reasonable to seek a global optimizer for the model. This process is facilitated if a smooth interpolating model is used. The major difficulty is that the interpolator is a model of the truth (the "truth" being the expensive analysis code) rather than the truth itself. Issues arise regarding refinement of existing models, and optimizing over sequences of models.

A minimalist approach is to run the expensive code only at the global optimizer of the model. However, in the likely event that this process does not yield a satisfactory stopping point, a more sophisticated strategy is required. Sequential modeling strategies are discussed below.

Since the interpolating models are generally simple formulas, it is reasonable to consider computation of the gradient and the Hessian of the model at any point in design space. These can be used to form quadratic model of the basins of interesting minimizers. It is also possible to infer the extents of model minimizer basins more directly by determining the coordinate distances from the local minimizer to some particular higher level set. The above tools can be used to determine regions in the locality of interesting model minimizers. More data points can be taken in these regions and used to form new interpolating models, applicable over the restricted regions. This "zoom-in" process can be repeated to produce interpolators of ever-increasing fidelity. In the terms of global optimization of the true problem, the "zoom-in" method is a purely "local" approach. That is, locally interesting regions are explored further, but the remainder of the design space is ignored.

A more "global" approach to the overall optimization problem is to obtain new data at points with the highest predicted modeling errors. This has the effect of encouraging global exploration, i.e. searching in regions with the potential for significant improvement.

In traditional global optimization it is usually best to balance local and global search. This is probably the case in sequential modeling as well. A simple method for balancing local and global concerns is to allocate some fraction of the new data points based on each criterion.

Related to the above approaches, is the idea of sequentially modeling over restricted regions, where the centers of the regions move in design space. This is essentially the *trust-region* concept generalized to interpolating models. Dennis and Torczon [12] have proven local convergence, in the absence of derivatives, for a trust-region based algorithmic framework, covering very general classes of models. Conn and Toint [13], and Powell [14] describe derivative-free, trust-region based modeling algorithms for local optimization. These methods could be used for local search, once global modeling has identified regions of interest.

6. CONCLUSIONS & AREAS OF RESEARCH

Optimization using interpolating modeling is a promising tool for global search on expensive functions. However, further research is required in several areas. One area, discussed above, is techniques and theory for optimization using sequences of models.

Another research area is dealing with constraint functions that are composed of outputs from the expensive code. Interpolating models can be computed for each constraint function, as well as for the objective function. However, the optimization process must account for infeasibilities due to discrepancies between the models of the constraints and the true constraints.

An additional research issue arises from the possibility that the analysis may be undefined, or the expensive analysis code may fail, in certain regions of design space. And, it may not be possible to predict the nature of these *undefined* regions. In fact, one potential use of DOE models is to characterize undefined regions.

References

[1] Owen, A. B. Orthogonal Arrays for Computer Experiments, Integration and Visualization, *Statistica Sinica*, 2, 439-452, 1992.

[2] Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D., Screening, Predicting, and Computer Experiments, *Technometrics*, 34, 15-25, 1992.

[3] Currin, C., Mitchell, T., Morris, M. and Ylvisaker, D., Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments, *Journal of the American Statistical Association*, 86, 953-963, 1991.

[4] Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., Design and Analysis of Computer Experiments, *Statistical Science*, 4, 409-435, 1989.

[5] Sacks, J., Schiller, S. B. and Welch, W. J., Designs for Computer Experiments, *Technometrics*, 31, 41-47, 1989.

[6] Wahba, G., *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics, 1990.

[7] Healy, M.J., Surface Fit/Optimization Methodology for Aircraft Engine Screening, The Boeing Company, D180-17884-1, January, 1975.

[8] Booker, A. J., DOE for Computer Output, Boeing Computer Services, BCSTECH-94-052, 1994.

[9] Efron, B. and Stein, C., The Jacknife Estimate of Variance, *The Annals of Statistics*, 9, 586-596, 1981.

[10] Johnson, M. E., Moore, L. M. and Ylvisaker, D., Minimax and Maximin Distance Designs, *Journal of Statistical Planning and Inference*, 26, 131-148, 1990.

[11] deBoor, C. and Ron, A., Computational Aspects of Polynomial Interpolation in Several Variables, *Mathematics of Computation*, 58, 705-727, 1992.

[12] Dennis, J. E. and Torczon, V., Managing Approximate models in Optimization Technical Report TR95-19, Rice University, June, 1995.

[13] Conn, A. R. and Toint, Ph. L., An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization, Preprint, 1995.

[14] Powell, M. J. D., A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation, in "Advances in Optimization and Numerical Analysis", Proceedings of the Sixth workshop on Optimization and Numerical Analysis, Oaxaca, Mexico", Volume 275 of *Mathematics and its Applications*, 51-67, Kluwer Academic Publishers, 1994.

****

BULLETIN BOARD

Announcement and Call for Papers

International Conference on Nonlinear Programming

(Beijing, Sept 2-5, 1996)

An *International Conference on Nonlinear Programming* will be held at the Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China, from September 2-5, 1996. It is organized by the Chinese Academy of Sciences and the Chinese Natural Science Foundation. Invited lectures on recent advances of nonlinear programming will be given. Invited Speakers (Preliminary list) will include:

J. Burke
A.R. Conn
J. Nocedál
M. Overton
Ph. Toint
R. Byrd
J. Moré
M.J.D. Powell
K. Tanabe
H. Wolkowitz
T. Coleman
L. Nazareth
R.B. Schnabel
R. Tapia
M.H. Wright

A limited number of short (20 minutes) papers will be accepted for presentation. Papers on theoretical, computational and practical aspects of nonlinear programming are welcome.

In part, this meeting is intended to honour the many contributions of Professor M.J.D. Powell to Optimization. It is hoped that this meeting will be similar to the one that Professor M.J.D. Powell organised in Cambridge in 1981. There will be no parallel sessions. Apart from the invited lectures and submitted short talks, there will also be discussion sessions. The conference proceedings will be published by an international publisher, and all the papers will be reviewed.

One or two sightseeing tours, including visiting the Great Wall, will be organized by the conference. There is also a possibility of organizing a post conference tour to Xi-an, an ancient capital of China, depending on the number of participants who are interested in such a trip.

Prospective participants (except invited speakers) should send their preregistration information giving address (postal and e-mail, if available) and accommodation preference(single or double bedroom in hotel) to the address below by post or e-mail. A further announcement will be sent to all those who preregister.

International Organizing Committee:

A.R. Conn, (IBM Watson Research Center, Yorks. USA)
J. Nocedál (Northwestern University, USA)
Ph. Toint (University of Namur, BELGIUM)
Y. Yuan (Chinese Academy of Sciences, CHINA)

Address for correspondence: For further information, please contact the following address or any of the International organizing committee.

Prof. Ya-xiang Yuan
State Lab. of Scientific and Engineering Computing
ICMSEC, Chinese Academy of Sciences
P.O. Box 2719, Beijing 100080, China
Tel: +86-10-255-9001, +86-10-254-5820
FAX: +86-10-254-2285
e-mail: yyx@lsec.cc.ac.cn

ERRATA IN ISSUE NO. 6 FEATURE

In the feature article of the previous issue, the surname of Stuart Kauffman was consistently misspelled as 'Kaufmann' (the Germanic form). The correct reference to his book is: Stuart A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, Oxford and New York, 1993. To make matters worse, there is yet a third way to spell this name, namely, 'Kaufman'. Mea culpa! - Ed.

SELECTED UPCOMING ARTICLES FOR SIAM J. OPTIMIZATION

Some Convergence Properties of the Modified Log Barrier Method for Linear Programming *M.J.D. Powell*
Fast Interior Point Methods for Bipartite Matching *Lov K. Grover*
Convergence of a Factorized Broyden-like Family for Nonlinear Least Squares Problems *Hiroshi Yabe and Naokazu Yamaki*
Sequential Quadratic Programming with Penalization of the Displacement *J. F. Bonnans and G. Launay*
Global Optimality Conditions and Their Geometric Interpretation for the Chemical and Phase Equilibrium Problem *Y. Jiang, W. R. Smith, and G. R. Chapman*
The Molecule Problem: Exploiting Structure in Global Optimization *Bruce Hendrickson*
An Information Global Optimization Algorithm with Local Tuning *Yaroslav D. Sergeyev*
Potential Transformation Methods for Large-Scale Global Optimization *Jack W. Rogers, Jr. and Robert A. Donnelly*

Existence and Regularity of Solutions to a Variational Problem of Mumford and Shah: A Constructive Approach *Yang Wang*

Augmented Lagrangian-SQP-Methods in Hilbert Spaces and Application to Control in the Coefficients Problems *Kazufumi Ito and Karl Kunisch*

Convex Analysis on the Hermitian Matrices *A. S. Lewis*

Classical Optimality Conditions Under Weaker Assumptions *Simon Di*

An Infinite-Dimensional Convergence Theory for Reduced SQP Methods in Hilbert Space *F.-S. Kupfer*

Dini Derivatives of the Marginal Function of a Non-Lipschitzian Program *D. E. Ward*

A Generalized Convexity and Variational Inequalities for Quasiconvex Minimization *Phan Thien Thach and Masakazu Kojima*

An Efficient Newton Barrier Method for Minimizing a Sum of Euclidean Norms *Knud D. Andersen*

On Long Step Path Following and SUMT for Linear and Quadratic Programming *Kurt M. Anstreicher*

A Superlinear Infeasible-Interior-Point Affine Scaling Algorithm for LCP *R. D. C. Monteiro and S. J. Wright*

On the Complexity of the Production-Transportation Problem *Dorit S. Hochbaum and Sung-Pil Hong*

On the Relationship Between the Curvature Integral and the Complexity of Path-Following Methods in Linear Programming *Gongyun Zhao*

The Mehrotra Predictor-Corrector Interior-Point Method as a Perturbed Composite Newton Method *R. Tapia, Y. Zhang, M. Saltzman, and A. Weiser*

An Infeasible Interior-Point Predictor-Corrector Algorithm for Linear Programming *Florian A. Potra*

Restricted Step and Levenberg-Marquardt Techniques in Proximal Bundle Methods for Nonconvex Nondifferentiable Optimization *Krzysztof C. Kiwiel*

REGARDING THE V&N

As mentioned in the Chair's column, a new editor will take over the reins following this issue. A periodic change is highly desirable in order to bring fresh input, direction, format (and perhaps even mode of distribution) to the views-and-newsletter. It has been a great pleasure to have been able to serve the optimization community in this capacity, and to have been given a free hand to define the V&N, during the past four years.

Larry Nazareth, Editor
Department of Pure and Applied Mathematics
Washington State University
Pullman, WA 99164-3113

email: nazareth@alpha.math.wsu.edu
or nazareth@amath.washington.edu

Fifth SIAM Conference on OPTIMIZATION

Victoria, British Columbia Canada

May 20–22, 1996
Victoria Conference Center

Sponsored by SIAM Activity Group on Optimization

ABOUT THE CONFERENCE

The field of optimization is a fascinating and lively blend of theoretical analysis, algorithm and software development, and scientific computing. This fifth conference will address the most important recent developments in linear, nonlinear, and discrete optimization. It will feature recent advances in optimization algorithms and software, as well as important applications of optimization in control, networks, manufacturing, chemical engineering, and operations research. An important emphasis of the meeting is also the increasing variety of connections between optimization and other fields of numerical analysis and scientific computing, such as differential equations. The organizers have made a particular effort to highlight some less traditional themes.

The conference will bring together mathematicians, operations researchers, computer scientists, engineers, and software developers. The wide scope of the conference should provide an excellent opportunity ror sharing ideas and problems among specialists in and users of optimization from academia, government and industry.

CONFERENCE THEMES

- Connections between continuous and discrete optimization
- Convex analysis and applications
- Differential-algebraic equations and their connections to optimization
- Derivative-free methods
- Industrial methodology and applications
- Nondifferentiable/structural optimization
- Semidefinite programming
- Stochastic programming

MINISYMPOSIA

A minisymposium is a two-hour session consisting of four presentations on a well-focused topic. Several sessions are being planned by the organizing committee. These will appear in future announcements in *SIAM News*.

ABOUT VICTORIA

Victoria is one of the top destinations in the Pacific Northwest. Mountains and sky, island and sea; everything meets here! Victoria offers an unforgettable experience for your next SIAM Conference — the Fifth SIAM Conference on Optimization. Do come and participate!

SIAM SHORT COURSES

Immediately preceding the conference – May 19, 1996

(offered simultaneously) on:

- Differential-Algebraic Equations and Their Connections to Optimization
- Optimization Software

INVITED PRESENTATIONS

Differential-algebraic Equations and Continuous Optimization
Uri M. Ascher, *University of British Columbia, Vancouver, Canada*

The Impact of Mathematics and Optimization at Boeing
Albert M. Erisman, *Boeing Information and Support Services, Seattle, Washington, USA*

Continuous Optimization Techniques in Discrete Optimization Problems
Martin Grötschel, *Konrad-Zuse-Zentrum, Berlin, Germany*

Convex Analysis and Applications
Adrian S. Lewis, *University of Waterloo, Waterloo, Canada*

Stochastic Programming
Andrzej Ruszczynski, *International Institute for Applied Systems Analysis, Vienna, Austria*

Direct Search Methods
Virginia Torczon, *College of William and Mary, Williamsburg, Virginia, USA*

Nondifferentiable/Structural Optimization
Jochem Zowe, *Universität Bayreuth, Bayreuth, Germany*

ORGANIZING COMMITTEE

Andrew R. Conn (Co-chair), *IBM T.J. Watson Research Center, Yorktown Heights, New York*

Margaret H. Wright (Co-chair), *AT&T Bell Laboratories, Murray Hill, New Jersey*

John T. Betts, *Boeing Computer Services, Seattle, Washington*

John R. Birge, *University of Michigan, Ann Arbor, Michigan*

Jonathan M. Borwein, *Simon Fraser University, Burnaby, Canada*

Georg H. Bock, *University of Heidelberg, Heidelberg, Germany*

Albert G. Buckley, *University of Victoria, Victoria, Canada*

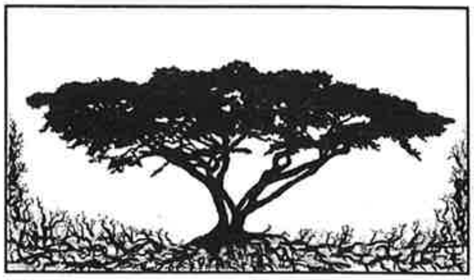
TO PARTICIPATE

The conference preliminary program and information on transportation, hotel, and registration will appear in the March 1996 issue of *SIAM News*. To receive your copy, contact *meetings@siam.org*.

Additional information is available through SIAM's World Wide Web site (*http://www.siam.org/conf.htm*) or by contacting SIAM, 3600 University City Science Center, Philadelphia, PA 19104-2688; Phone 215-382-9800; Fax 215-386-7999.

Iterative Methods for Linear and Nonlinear Equations

C. T. Kelley



Frontiers in Applied Mathematics 16

1995
xiii + 165 pages
Softcover
ISBN 0-89871-352-8
List Price $32.50
SIAM Member Price $26.00
Order Code FR16

Linear and nonlinear systems of equations are the basis for many, if not most, of the models of phenomena in science and engineering, and their efficient numerical solution is critical to progress in these areas. This is the first book to be published on nonlinear equations since the mid-1980s. Although it stresses recent developments in this area, such as Newton-Krylov methods, considerable material on linear equations has been incorporated. This book focuses on a small number of methods and treats them in depth.

The author provides a complete analysis of the conjugate gradient and generalized minimum residual iterations as well as recent advances including Newton-Krylov methods, incorporation of inexactness and noise into the analysis, new proofs and implementations of Broyden's method, and globalization of inexact Newton methods.

Examples, methods, and algorithmic choices are based on applications to infinite dimensional problems such as partial differential equations and integral equations. The analysis and proof techniques are constructed with the infinite dimensional setting in mind and the computational examples and exercises are based on the MATLAB environment.

Contents

Preface; How to Get the Software; **Part I: Linear Equations.** Chapter 1: *Basic Concepts and Stationary Iterative Methods;* Chapter 2: *Conjugate Gradient Iteration;* Chapter 3: *GMRES Iteration;* **Part II: Nonlinear Equations.** Chapter 4: *Basic Concepts and Fixed Point Iteration;* Chapter 5: *Newton's Method;* Chapter 6: *Inexact Newton Methods;* Chapter 7: *Broyden's Method;* Chapter 8: *Global Convergence;* Bibliography; Index.

Audience

This book can be used as a tutorial and a reference by anyone who needs to solve nonlinear systems of equations or large linear systems. It may also be used as a textbook for introductory courses in nonlinear equations or iterative methods or as source material for an introductory course in numerical analysis at the graduate level. The reader should be familiar with elementary numerical analysis, linear algebra, and the central ideas of direct methods for the numerical solution of dense linear systems.

About the Author

Professor Kelley has an appointment in the Department of Mathematics and is a member of the Center for Research in Scientific Computation at North Carolina State University. He is the current vice-chair of the SIAM Activity Group on Optimization and serves on the editorial board of the *SIAM Journal on Optimization.*

Complete tables of contents and additional information about SIAM publications are available through the World Wide Web *(http://www.siam.org)* or via SIAM's Gopher server *(gopher.siam.org).*

To ORDER

Use your credit card
(AMEX, MasterCard, and VISA):
Call toll free in USA: 800-447-SIAM
Outside USA call: 215-382-9800
Fax: 215-386-7999; E-mail: service@siam.org

Or send check or money order to:
SIAM, Dept. BKAG95, P.O. Box 7260
Philadelphia, PA 19101-7260

Payments may be made by wire transfer to SIAM's bank:
PNC Bank, 3535 Market Street, Philadelphia, PA 19104; ABA Routing #031000053; Account Name: Society for Industrial and Applied Mathematics; Account #8550970454

Shipping and Handling
USA: Add $2.75 for the first book and $.50 for each additional book.
Canada: Add $4.50 for the first book and $1.50 for each additional book.
Outside USA/Canada: Add $4.50 per book.
All overseas delivery is via airmail.

siam®

Science and Industry
Advance with Mathematics