SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

No.4 Spring 1994

OPTIMIZATION SOFTWARE FOR NONSMOOTH & STOCHASTIC PROBLEMS: A MISCELLANY

The recently published *Optimization Software Guide* (SIAM, 1993) of Jorge Moré and Stephen Wright surveys 75 major packages and libraries in areas where the greatest progress, to date, has been made in developing optimization software: unconstrained optimization; nonlinear least-squares; nonlinear equations; LP; QP; bound-constrained; constrained optimization; network optimization; integer programming. The topics of nonsmooth optimization and optimization under uncertainty (stochastic programming) are relegated to the short concluding chapter on Miscellaneous Problems, because software in these areas is only just beginning to reach the same level.

Here are a few quotations ('verbal'; "printed") that characterize the latter two subjects:

> 'I regard optimization under uncertainty as the most fundamental problem of decision science' - G.B. Dantzig.
>
> 'It is only a slight exaggeration to say that 90 percent of applications in nonsmooth optimization arise via decomposition in one form or another, and the remaining 10 percent via eigenvalue computations' - C. Lemaréchal
>
> "In fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity"- R.T. Rockafellar
>
> "In *principle*, the decentralization of decisions with observance of the total objective value of the problem is possible by means of the correct construction of objectives in submodels. We point out here the brilliant mathematical formalism of the idea of decomposition by G. Dantzig and P. Wolfe. The value of their paper of 1960 is far a greater than the *limits of the algorithm* they proposed and its mathematical foundation. It gave rise to discussions and *alternative treatments* all over the world. - L. Kantorovich. (italics ours)

In particular, these quotations point to the central role played by convex analysis and the decomposition principle. (Note that it is important to distinguish between the broad and far-reaching D-W decomposition principle on the one hand, and the *particular* D-W algorithm with its well-known computational drawbacks on the other, as highlighted in the last quotation). An increased reliance on decomposition also goes hand-in-hand with increased opportunities for exploiting parallelism.

The task of developing software is made especially challenging by the above considerations, the variety of theoretical models from which to choose, and the fact that the two subjects are still very much on the research frontier. Practical nonsmooth and stochastic problems lend themselves more readily to the development of *specially-tailored* models and solution techniques in a high-level, user-extensible language, for example, Matlab (with appropriate toolboxes). Moreover, the current shortage of robust, *general-purpose* nonsmooth &/or stochastic optimization software makes practitioners understandably reluctant to formulate practical models of this type. The lack of availability of practical models, in turn, results in there being less incentive to engineer user-oriented, robust software – a classic Catch-22 situation!

Despite the difficulties, there has been considerable progress. Advanced implementations in the area of nonsmooth optimization include: BTC and BTNC (H. Schramm and J. Zowe, U. Bayreuth, Germany; C=convex, NC=nonconvex); M1FC1 (C. Lemaréchal; INRIA, France); NOA 3.0 (K. Kiwiel, Systems Research Institute, Warsaw). All three packages are based on the bundle concept. Advanced implementations that are available or under development for optimization under uncertainty include: SP/OSL (A. King, IBM, Yorktown Heights; multistage with scenarios); SQG (A. Gaivoronski, ITALTEL, Milan; general expectation problems with linear/nonlinear constraints); MSLiP (H. Gassmann, Dalhousie U., Canada; multistage recourse problems); DECIS (G. Infanger, SOL, Stanford U.; two-stage recourse); SLP-IOR (P. Kall/J. Mayer; U. Zurich; two-stage recourse and chance constraints).

Thus, when a future edition of the Optimization Software Guide is begun (as anticipated by its authors), it is likely that the fields of nonsmooth and stochastic optimization will have advanced to a point warranting a much broader inclusion.

Contents:

CHAIRMAN'S COLUMN

by A.R. Conn

Our SIAM activity group is intended to be of interest to a broad range of optimizers. Most of my remarks below are made particularly with continuous nonlinear programmers in mind, but I would be surprised if the comments applied only to us. Thus, I apologize for the narrowness (but in the same breath, withdraw the apology). Most of us, I assume, agree that the world is not linear and so one might assume that nonlinear programmers have had a great deal of success in "selling their (soft)wares in the marketplace". My impression is that in fact, especially if one is knowledgeable about the current state of the art of nonlinear programming, we have been remarkably unsuccessful at convincing practitioners to use nonlinear models and sophisticated software. Thus the point of this column is to ask why we have failed so miserably in this regard and what we should do about it. As you can see, I do not raise the possibility that I am mistaken in my conviction that we are failing in this respect. If someone could convince me otherwise I would be delighted, but meanwhile I will comment on the situation as it presently appears to me.

Firstly, I think that the *major* problem is that nonlinear optimization requires sophisticated users. I do not wish to imply that linear programming cannot benefit from an intelligent and knowledgeable user, but even a naive user is likely to be able to solve difficult linear programming problems using provided software as a "black box". In order to expect to do the same for nonlinear problems, the user has to be not only naive but myopic or perhaps even blind. For instance, suppose one has a rather complicated nonlinear optimization problem that comes from a genuine problem in manufacturing. A not unusual scenario is that the problem is deemed sufficiently complicated that the function values are determined via some simulation package. Thus one can be almost certain that no derivatives are provided. If the application is endeavouring to make use of what has filtered down as the latest optimization technology, it is likely to use something like sequential quadratic programming, provided by the writers of the simulation package, for example. First derivatives will be provided by finite differencing. If the package is an average one and the problem is significantly nonlinear and involves more than a dozen variables, what do you think are the chances of this black box approach being successful? Very small, in my estimation. Thus the most likely result is frustration on the part of the user and a feeling that nonlinear optimization is useless. Ideal encouragement for them to return to their linear models, with which they had great success, since the linear programming codes almost always worked for them.

I mentioned myopic and/or blind above, because there is another situation. Look at almost any issue of, for example, OR/MS Today, and you will find what I call the 'blind optimist/opportunist' advertisement.

Buy SLOPTimize — for less than $1000 we can provide you with software that does automatic scaling in a unique and rigorous way, thus enabling it to reliably solve YOUR mixed integer linear and nonlinear optimization problems. No derivatives required or encouraged. Particularly recommended for problems in less than one thousand variables. Others make the claims, we have the results!

Such packages not only do not require sophisticated users, they don't want them. They provide answers but not correct ones. In a typical instance they might be using, for example, the SIMPLEX approach of Nelder and Mead or an approach that is very much based upon steepest descent. My point here is not to blame the providers of such packages, but rather to indicate that it is the failure of us to consistently convince practitioners that we can do better, that enables such packages to survive.

However, it is not easy to know how to best address the problem. I consider myself to be a sophisticated nonlinear programmer. I have access to a number of the leading nonlinear optimization packages but nevertheless, it is often difficult to solve an application problem. It is not unusual to have the original problem badly posed, badly scaled and containing errors. The same, of course, can be true for linear programming, but it is significantly less likely. I am of the opinion that, for the foreseeable future, applied nonlinear programming is likely to require, to be successful, a degree of sophistication that will be unreasonable to expect from even ten percent of the practitioners who can benefit enormously from what nonlinear optimization has to offer. Thus I feel that it is essential that we begin collecting and advertising well-documented success stories, to promote the idea that one can and should solve nonlinear models. One problem is the undoubtedly proprietary aspects of most applications. If a petroleum company has great success in finding oil-fields because it is using sequential quadratic programming techniques, it is not likely to tell its rivals to stop using PARTAN. However, we could make a start by having those of us who work in nonlinear optimization make a real effort to tackle useful problems and in so doing, attempt to broadcast the fact that one can successfully solve such problems. Ideally, one should at the same time make it clear, in the instances in question, why linear models would have been inadequate. I realise that this approach is hardly revolutionary but meanwhile the (slightly exaggerated) status quo in the real world is either a belief that black box nonlinear optimization works (and I do not think it does) or that all optimization has of necessity to be linear.

As always, I welcome your opinion. My email address is arconn@watson.ibm.com

FORUM ESSAYS

INVITED AUTHOR'S PREVIEW: CONVEX ANALYSIS AND MINIMIZATION ALGORITHMS

by Claude Lemaréchal [1]

It is a cliché to say that convex analysis is a relevant subject for optimization. Convexity has always been known to be useful, for example, for establishing the equivalence between critical points and local-global minima, for the existence of a minimum point in infinite-dimensional spaces, and so on.

The more recent emergence of problems and methods for nonsmooth optimization increased the importance of convexity. The latter subject was taken beyond the framework of optimization theory, and began to play a role in the development and analysis of optimization *algorithms*. A new field of "convex numerics" was born, and its domain of application now includes interior-point methods (see the recent *Interior Point Polynomial Algorithms in Convex Programming* by Yu. Nesterov and A.S. Nemirovski, SIAM Studies in Applied Mathematics 13, 1993).

However, it is fair to say that convex analysis is not a subject generally mastered by the optimization community (and, incidentally, by many other communities). Indeed, it is not easy to become knowledgeable on the matter. At least for the "nonlinear" part of us, we have grown accustomed to manipulating Taylor developments and similar tools of ordinary calculus. Convex analysis demands a fresh brain: most manipulations deal with inequalities, subspaces are replaced by cones, differentiation no longer results in linear mappings ....

Another plain truth is that there is only one path to expertise in convex analysis: the "holy" book of R.T. Rockafellar (*Convex Analysis*, Princeton University Press, 1970). Its unmatchable qualities: exhaustivity, clarity, mathematical elegance ... have made it *the* reference for more than twenty years, and it will certainly remain so for many more decades. Nevertheless, it is quite difficult to read, because of its encyclopaedic character. A key is needed to unlock this castle, which has sometimes been called "probably the only American Bourbaki-work". We believe that this explains why convex analysis is still confined to a private club: it cannot easily be taught in an applied mathematics context, on the basis of the holy book.

The reason for this preamble is the following piece of self-advertisement. Springer-Verlag has just issued in the Grundlehren (the "Yellow Series", founded by R. Courant in 1921) a book in two volumes entitled *Convex Analysis and Minimization Algorithms*, by J.-B. Hiriart-Urruty and C. Lemaréchal. The aim of this work is twofold:

- To serve as a textbook on convex analysis, in which pedagogy would be the keyword. Part I, subtitled *Fundamentals*, is mostly devoted to this task.
- To give an account of nonsmooth optimization in book form, which is the main content of Part II: *Advanced Theory and Bundle Methods* (sorry if we limited ourselves to these methods, which are also the limits of our competence in the field!).

It is for its pedagogical merit that we have chosen such a bi-disciplinary approach, in which the applications (minimization algorithms) serve to motivate and illustrate the theory (convex analysis).

Pedagogy also urged us to call on geometric intuition (there are 177 figures), possibly at some sacrifice of mathematical elegance. This book can be qualified as definitely un-Bourbaki; note, however, that mathematical rigor is hopefully still present – after all, we are both French!

Again for pedagogical purposes, our development is very progressive, sometimes even repetitive; it evokes a spiral rather than a straight line. In addition to being redundant, we are often digressive: to explain a delicate point, to give examples, or to link together remote concepts. All this explains the 763 pages making up the two parts: the same job could probably be done in about half the amount, but this would kill the *raison d'être* for the book.

To summarize our mentality when writing this book, let us say that we will be happy if whoever has read it can then embark on reading Rockafellar.

A few comments on some of our options will illustrate the above points.

– The first part has 6 chapters on convex analysis, and 2 on numerical algorithms. In Part II these numbers become 2 and 5 respectively. In other words, theory and applications are interspersed all the way.

– Chapter I studies convex functions of one real variable rather deeply (for example, it includes the conjugacy operation). Needless to say, most of its material is repeated in later chapters, but in a more advanced setting.

– Chapter II is an introduction to optimization algorithms, with a large part of it (1/3) devoted to linesearches. Thus, the bi-disciplinary aspect is readily settled after the first two chapters.

[1] INRIA, Rocquencourt, B.P. 105, F–78153 Le Chesnay, France. e-mail: Claude.Lemarechal@inria.fr

– A full chapter (Chap. V) is devoted to sublinear functions, i.e. convex functions that are positively homogeneous. They are isomorphic to closed convex sets via the supporting operation, and this generalizes the isomorphism between vectors and linear forms. This fact, a foundation stone of convex analysis, is not much developed elsewhere.

– The subdifferential is the most advanced concept appearing in the first part. Yet, it is limited to the case of finite-valued functions, so that geometric intuition can fully play its role. We give a number of equivalent definitions: starting from directional derivatives, from supporting slopes, from normal cones to the epigraph; these are fairly classical but the last definition, starting from limits of gradients, is not. Its merit is to be very intuitive, especially in the context of numerical algorithms; furthermore it directly suggests the nonconvex generalizations à la Clarke (*Optimization and Nonsmooth Analysis*, SIAM Classics in Applied Mathematics 5, 1990).

– Chapter XI studies the approximate subdifferential, which serves our purposes from three points of view: it allows at the same time the study of the ordinary subdifferential in the general case, it is of importance for bundle methods, and it gives some insight into the conjugacy operation.

– All along our development, we insist on calculus rules. When functions are combined, classical calculus studies the corresponding combination of their derivatives. In convex analysis, there are several fruitful correspondences: between sublinear functions and closed convex sets, via the supporting operation; between pairs of convex functions, via the conjugacy operation; between convex functions and closed convex sets, via the subdifferential (possibly approximate). In each case, we give a systematic review of calculus rules, enabling a whole set of algebraic manipulations. These reviews are sometimes tedious, but they are also essential if one wants to become technically skillful in the matter.

– Our approach to duality in constrained optimization is totally oriented towards dual algorithms (Lagrangian relaxation). The corresponding chapter (Chap. XII) is therefore viewed as a numerical one. Furthermore, we have given to convexity the least possible role in our development. This makes things much clearer, and combinatorial problems are thus accepted by the theory. Indeed Lagrangian relaxation lies at the intersection between nonlinear and combinatorial optimization; through it, the two fields have a chance to benefit better from each other.

– Bundle methods cover a total of four chapters, extending from Chap. IX to XV. This is a lot, especially as they are limited to the simplest situation of minimizing a convex function without constraints. Actually, only Chap. XV is crucial; the other three could have been placed after it, as they rather give additional and less important material. We chose the present order to struggle against human laziness: external references from inside Chap. XV are not totally avoided, so that the reader is forced to get some knowledge of the other chapters, possibly less important but just as informative.

A final word: it took us seven years to write this book. We thought our task deserved them, in the hope that the result could become useful for the purpose of learning and teaching. Convex analysis is rich enough to become part of the academic curriculum: it is a good educational tool, and has applications in several scientific domains such as mathematical economics, mechanics, thermodynamics, approximation theory, statistics and statistical mechanics..., and may it bring fresh blood to numerical optimization as well!

****

STOCHASTIC PROGRAMMING

by Alan J. King[1] and Roger J-B Wets[2]

1. INTRODUCTION

There are few practical optimization problems where the modeler is not faced with uncertainty about the value to assign to some of the parameters. The source of the uncertainty can be the lack of reliable data, measurement errors, or uncertainty about future, or unobserved, events; there may even be uncertainty about the structure of the problem itself.

In some instances no harm will come from ignoring these uncertainties. One may rely on "best estimates" and, if needed, follow up with post-optimality parametric analysis. But there are quite a number of situations when proceeding in this manner produces "solutions" whose implementation could lead to disaster! For example, designing a master production plan without taking into account the inherent uncertainty about future markets leaves the manufacturer exposed to large losses if the evolution of the market doesn't nearly match the predictions. A valid approach would account for a certain distribution of future sales, technological developments, commodity prices, etc.

There are a number of ways of handling uncertainty. One that has proved useful in a variety of situations is to assign to the uncertain parameters a probability distribution (based on statistical evidence or not), design

[1]IBM Research Division, Yorktown Heights, NY. e-mail: kingaj@watson.ibm.com

[2]Department of Mathematics, University of California, Davis, CA. e-mail: rjbwets@ucdavis.edu

"recourse" functions that model the risk if certain goals or targets are missed, and optimize the expected value of the recourse. This would cast the optimization model as a *stochastic programming problem.*

Stochastic programming is concerned with practical procedures for decision-making under uncertainty, by modeling uncertainties and risks associated with decisions in a form suitable for optimization, and devising approximation and decomposition methods for computing solutions.

The subject matter of stochastic programming is shared by other fields. Statistical decision theory is concerned with the processing of sequential observations to make decisions (for example, whether or not to repair a certain machine). Stochastic dynamic programming, or stochastic control, is concerned with the computation of feedback, or control, laws that specify optimal actions based on the state of the system (for example, at what level of inventory should one place an order to restock). The scope of any of these fields can be generalized to cover every aspect of decision-making under uncertainty. But practical aspects of mathematical analysis and computation in this challenging subject lead to tangible differences between approaches. Articles illustrating the broad range of mathematical treatments of decision-making under uncertainty can be found in Dempster [3] and Ziemba & Vickson [11].

Models of uncertainty and risk that include enough detail to be useful in industrial, business, or government planning will generate problems that are impossible to solve because of the exponential explosion of states. For instance, a stochastic process with only four possible realizations in each of ten time periods generates over one million sample paths! The approach employed by stochastic programming is to focus the attention on the first stage decision—the "now"-decision that hedges against future uncertainty. When aspects of the uncertainty in the problem become known, "recourse decisions" responding to the new information may be made.

For example, in an investment problem, the uncertainty in asset prices might be modeled by diffusion processes. The investor might select a target performance level and formulate the risk in the problem as the total expected shortfall of the portfolio value below the target, net of taxes and transaction costs, over the next ten quarters. The first-stage decision is the initial allocation of available funds to assets, and the recourse decisions are the proportion of the portfolio bought or sold at each turn of the quarter.

A simplistic approach to the problem of modeling uncertainty that should be mentioned here is that of "scenario analysis". This practice is common in business and industry. One produces a number of simulations, say, of asset prices over the next ten quarters, then one views the impact of a decision policy, like a fixed allocation between long and short term bonds, by examining the outcome of the policy under each simulation. Through a process of exhaustive search one looks for a policy with a reasonable distribution of outcomes. While this procedure offers a simple way to incorporate uncertainty into a decision model, it is not operationally sound. It cannot find, unless through exceptional luck, a decision that hedges against uncertainty.

2. MATHEMATICAL STRUCTURES

The mathematical investigation of stochastic programming combines the subjects of probability theory, statistics, non-smooth analysis, and linear programming. To illustrate the mathematical structure, we outline the construction of a multistage stochastic linear program; for a more complete presentation, including non-linear formulations, see the introduction to Ermoliev & Wets [5]. We first describe the dynamic structure of the problem as a multistage linear program, and then introduce the stochastics.

The first stage decision is a linear program (ignoring subsequent stages):

$$\text{minimize } c_0 x$$

$$\text{s.t. } A_0 x \geq b_0.$$

In each subsequent stage from time stage 1 until the ending time stage T, we make recourse decisions, also modeled as linear programs, which depend on decisions previously made. The recourse linear program is:

$$\text{minimize}_{y_t} \; c_t y_t$$

$$\text{s.t. } A_{tt} y_t \geq b_t - (A_{t0} x + \cdots + A_{t,t-1} y_{t-1}).$$

It is important to note that the inequalities in the recourse linear program include all interperiod dynamical relationships and intraperiod constraints.

In the general stochastic programming model, the stochastic process involves every coefficient of the objective, right-hand side, and matrix, in the recourse linear programs; although in practice only a few such coefficients will be random. The distribution of coefficients in any time stage will in general depend on the history of the stochastic process up to that point. The subscript "t" appended to the expectation operator "E" will denote expectation conditioned on the history of the process up to (but not including) stage t. Now define, in recursive fashion, the value, or cost-to-go, functions. The value function depends on the stochastic process; we signal this dependence by including the Greek letter "omega" in the argument list. }

$$f_{t-1}(x, y_{t-1}, \ldots, y_1; \omega) = \min_{y_t} c_t y_t + E_t \{ f_t(x, y_t, \ldots, y_1; \omega) \}$$

$$\text{s.t. } A_{t0} x + A_{t1} y_1 + \cdots + A_{tt} y_t \geq b_t.$$

This definition begins at stage T (where the right side has no value-function term in the objective) and proceeds until stage 1 (where the left side has no dependence on any recourse decision). The stochastic programming problem can now be described in terms of the first stage decision variables alone:

$$\text{minimize}_x \quad c_0 x + E\{f_0(x;\omega)\}$$
$$\text{s.t.} \quad A_0 x \geq b_0. \tag{1}$$

The mathematics of stochastic programming is devoted to understanding the formulation in (1). Two noteworthy aspects distinguish it from other studies in nonlinear programming or classical probability and statistics. First, the objective function is defined by an integral. Second, the integrand is the value function of an optimization problem, which would therefore not generally possess a first derivative in the classical sense nor even be finite-valued. In general, it is hopeless to try to find a closed-form representation of the integral as a function of the first-stage decision (except possibly in very simple cases). Nevertheless, this formulation is amenable to analysis and computation. The basic tools are those of probability theory, and variational analysis. This is an area of mathematics whose theoretical challenge is matched only by the practical importance of the applications themselves.

3. HISTORY, MODELS AND COMPUTATION

The history of stochastic programming closely follows that of the development of sophisticated optimization algorithms and ever more powerful computers.

One of the earliest "stochastic programs" was H.M. Markowitz's mean/variance formulation of the portfolio optimization problem: minimize variance of return subject to a constraint on expected return. This amounts to selecting a target return and minimizing the expected value of a (quadratic) recourse function that penalizes the difference between portfolio return and target. The resulting objective is quadratic in the decision variables, and can be solved by a version of the simplex method. Chance constrained problems, where the probability of a bad event (e.g. bridge collapse) is constrained, are in wide use in engineering and power systems design. A. Prekopa showed that constraints specifying the probability that a random vector be coordinate-wise less than or equal to a given problem variable reduce to nonlinear convex constraints in the variable when the probability distributions involved are log-concave; see Ermoliev & Wets [5] for articles relating to chance constraints. These two formulations are popular because components of the probability distribution (e.g. variance) can be incorporated directly into the optimization model, leading to low-dimensional problems with some hope of solution.

The more general subject of stochastic programming outlined above, or linear programming under uncertainty as it was called then, was independently introduced by G.B. Dantzig and E.M.L. Beale in 1955. Here, an explicit discrete description of the sample space is introduced either at the outset, or as part of an algorithmic procedure. This model is capable of representing a great variety of practical decision problems through various modeling devices of linear programming. The drawback of the general model of linear programming under uncertainty is the curse of dimensionality. Unless one can be clever, or lucky, one is faced with solving a problem with millions/billions of variables and constraints.

The challenge of solving such problems has led to many interesting computational and theoretical developments. Chief among these are the L-shaped method of Van Slyke & Wets [10] and its multistage extension, Birge [1], which partitions the stochastic program by decision stage, and the theory of epi-convergence (whose development was partially motivated by its applications in stochastic programming) that justifies sampling and other methods of approximating the integration—see King & Wets [6], for example. Two recent developments point to exciting prospects for the solution of these problems: the aggregation method which decomposes by information field (Rockafellar & Wets [8]) and importance sampling in the L-shaped method (Dantzig & Glynn [2]). Decomposition permits very large problems to be solved on multiple processors. Sampling is used to represent the information in the uncertainty model with just a few data points. These two ideas, decomposition and approximation, are the keys to computational progress in stochastic programming.

Until recently, few stochastic programming applications could be formulated and successfully solved. Some exceptional recent efforts in two-stage stochastic programming are bank asset-liability management (Kusy & Ziemba [7]) lake pollution management (Somlódy & Wets [9]), and manufacturing capacity expansion (Eppen, Martin & Schrage [4]). Today, due to the explosion in power and capability of computers and optimization algorithms, multistage stochastic programming formulations are emerging from academia and the leading optimization laboratories into operational applications. The major application areas include financial asset/liability management over multi-year horizons, multistage production planning models with uncertain demand, power systems management over multiple time periods, forest harvest management, and long-range energy-economic planning models.

References

[1] Birge, John R., 1985. "Decomposition and partitioning methods for multi-stage stochastic linear programs", *Operations Research* 33: 989–1007.

[2] Dantzig, George B. and Peter W. Glynn, 1990. "Parallel processors for planning under uncertainty", *Annals of Oper. Res.* 22: 1–21.

[3] Dempster, Michael A.H., 1980. *Stochastic Programming.* New York: Academic Press.

[4] Eppen, Gary D., R. Kipp Martin and Linus E. Schrage, 1989. "A scenario approach to capacity planning", *Operations Research* 37: 517–527.

[5] Ermoliev, Yuri and Roger J-B Wets, 1988. *Numerical Techniques for Stochastic Optimization.* New York: Springer-Verlag.

[6] King, Alan J. and Roger J-B Wets, 1991. "Epi-consistency of convex stochastic programs", *Stochastics* 34: 83–92.

[7] Kusy, M.I. and William T. Ziemba, 1986. "A bank asset and liability management model", *Operations Research* 34: 356-376.

[8] Rockafellar, R. Tyrrell and Roger J-B Wets, 1991. "Scenarios and policy aggregation in optimization under uncertainty", *Math. of Oper. Res.* 16: 119–147.

[9] Somlyódy, László and Roger J-B Wets, 1988. "Stochastic optimization models for lake eutrophication management", *Operations Research* 36: 660–681.

[10] Van Slyke, Richard M. and Roger J-B Wets, 1969. "L-shaped linear programs with application to optimal control and stochastic programming", *SIAM J. Appl. Math.* 17: 638–663.

[11] Ziemba, William T. and R.G. Vickson, 1975. *Stochastic Optimization Models in Finance.* New York: Academic Press.

****

STOCHASTIC QUASIGRADIENT METHODS

by Yury Ermoliev[1] and Alexei A.Gaivoronski[2]

1. INTRODUCTION

Stochastic Quasigradient (SQG) methods constitute a family of iterative stochastic optimization procedures. Their main distinctive feature is the utilization of random directions for choosing the next approximation to the optimal solution. Because a deterministic process can be viewed as a special case of a stochastic process, one can expect SQG techniques to provide additional flexibility when used to solve well-known problems, as well as new opportunities for addressing challenging classes of problems that cannot be solved by conventional deterministic approaches.

There are at least three main application areas for SQG methods:

[1]IIASA, Laxenburg, Austria

[2]ITALTEL and Milan University, Milan, Italy, e-mail: gaivoron@settimo.italtel.it

1. Deterministic problems involving difficulties in the calculation of deterministic descent directions, like finite-difference approximations of gradients (large-scale, distributed, non-stationary optimization).

2. Multiextremal problems where it is important to by-pass locally optimal solutions (both smooth problems and problems with discontinuities resulting, in particular, from the existence of discrete variables).

3. Stochastic optimization problems involving uncertainties in evaluation of objective and constraints functions, including learning, estimation and adaptation.

In this article, we give a brief introduction to SQG methods. Many important issues have been deliberatly left aside, for example, software and related topics. Good sources for further reading are the collection of papers of co-authors in *Numerical Techniques for Stochastic Optimization,* Yu. M. Ermoliev and R. J-B. Wets eds., Springer, 1988 and the Proceedings of triennial conferences on stochastic optimization published in *Annals of Operations Research*, where one can also find further references.

2. GENERAL IDEA

We consider the problem

$$\min_{x \in X} F^0(x), \quad F^i(x) \leq 0, \quad i = 1:m \tag{2}$$

where the set X has a "simple" structure (for example, defined by linear constraints). In what follows we will define $F(x) = (F^0(x), ..., F^m(x))$ and refer to $F(x)$ as the *problem function.* The main idea is to use statistical (biased or unbiased) estimates of objective and constraints functions and/or their gradients instead of exact values of these functions. In other words, a sequence of approximate solutions $x^0, x^1, ..., x^s, ...$ is constructed by using random vectors h_s, ξ^s such that

$$|\mathbf{E}\{h_s|\mathbf{B}_s\} - F(x^s)| \to 0 \tag{3}$$

$$\| \mathbf{E}\{\xi^s|\mathbf{B}_s\} - F_x(x^s) \| \to 0 \tag{4}$$

where $F(x^s)$ is a problem function and $\mathbf{B}_s$ is a σ-field generated by the history of the process. In many cases requirements (3)-(4) can be relaxed. Vectors ξ^s satisfying (3)-(4) are called *stochastic quasigradients.*

These vectors are used in the following iterative SQG scheme:

$$x^{s+1} = \Phi(x^s - \rho_s \Psi(\bar{\xi}^s, \bar{h}_s)) \tag{5}$$

where $\bar{\xi}^s = (\xi^1, ..., \xi^s)$, $\bar{h}_s = (h_1, ..., h_s)$ and $\Psi(\cdot)$ transforms the set of quasigradients obtained up to iteration s into the current step direction. For example, $\Psi(\bar{\xi}^s, \bar{h}_s) = \xi^s$; other alternatives include the moving average of the previous quasigradients:

$$\Psi(\bar{\xi}^s, \bar{h}_s) = (1 - \alpha_s) z^s + \alpha_s \xi^s$$

or stochastic quasi-Newton methods

$$\Psi(\bar{\xi}^s, \bar{h}_s) = A_s \xi^s$$

where the matrix A_s reflects the second order information gathered through processing of the sequences $\bar{\xi}^{s-1}, \bar{h}_{s-1}$

Furthermore, ρ_s is the stepsize which can be scalar or vector and can reflect information about the process gathered on previous iterations. Operator $\Phi(\cdot)$ processes constraints and assures that the limit points of the sequence $x^1, ..., x^s$ are feasible. In the simplest case when the feasible set X is defined by linear constraints, this operator can be a projection on the set X. More complicated constraints can be processed by successive linearization techniques or using penalty and/or Lagrangian approaches. Thus, the simplest stochastic quasigradient method for the case when $m = 0$ has the form:

$$x^{s+1} = \Pi_X(x^s - \rho_s \xi^s) \tag{6}$$

where $\Pi_X(z)$ projects the point z on X.

3. CONVERGENCE RESULTS

The convergence, with probability 1, of the sequence $\{x^s\}$ to the solution set X^* has been proved for nondifferentiable (weakly convex, locally Lipshitz and even semicontinious) functions. The proofs rely on the existence of Liapunov-type functions which are decreased in a non-monotonic manner along the trajectory of the approximate solutions. Weaker notions of convergence have also been studied.

4. EXAMPLES OF THE STOCHASTIC QUASIGRADIENT APPROACH

Large-scale optimization: Suppose that $F(x_1, ..., x_n)$ is a differentiable function for which the calculation of finite differences is time-consuming because it needs at least $n+1$ function evaluations, i.e. $F(x)$ may be defined on solutions of differential equations or n may be very large. The stochastic vector

$$\xi^s = \sum_{i=1}^{M} \frac{F(x^s + \Delta_s h^i) - F(x^s)}{\Delta_s} h^i, \tag{7}$$

where $M \geq 1$ and h^i has independent uniformly distributed on $[-1, 1]$ components, satisfies (4). The calculation of ξ^s can require as little as two function evaluations independently of the dimensionality of the problem.

Global optimization: The simplest way to introduce "inertia" in a gradient-type method in order to bypass local solutions is to perturb the gradient by a random vector w^s, i.e., instead of $F_x(x^s)$, consider the vector

$$\xi^s = F_x(x^s) + w^s, \quad \mathbf{E}w^s = 0.$$

A special choice of w^s corresponds to simulated annealing. A different perturbation is useful for nondifferentiable functions:

$$\xi^s = F_x(\bar{x}^s), \quad \bar{x}^s = x^s + w^s \tag{8}$$

where the random vector w^s has a density and, with probability 1, $\| w^s \| \to 0$ for $s \to \infty$.

Nondifferentiable and discontinuous optimization: In this case the subgradient algorithm that uses finite difference approximations does not converge. The slight change to the random vectors

$$\sum_{i=1}^{n} \frac{F(\bar{x}^s + \Delta_s e^i) - F(\bar{x}^s)}{\Delta_s} e^i \tag{9}$$

where $\bar{x}^s$ is defined in (8), ensures the convergence of gradient-type procedures even for locally Lipshitz functions.

Minimax problems: Here it is difficult to evaluate exact values of the objective function

$$F(x) = \max_{y \in Y} f(x, y)$$

especially when the inner maximization is a nonlinear optimization problem of general type. The SQG approach for minimization of $F(x)$ is to use the random vector

$$\xi^s = f_x(x^s, y^s), \quad y^s = \arg\max_{y \in Y^s} f(x^s, y)$$

where $Y^{s+1} = y^s \cup z^s$ and z^s is a random vector distributed with nonzero density on the set Y.

Stochastic programming and optimization of stochastic systems: A rather general stochastic optimization model has the following objective/constraint functions

$$F(x) = \int f(x, \omega) P(x, d\omega) \tag{10}$$

where the probability measure $P(\cdot, d\omega)$ may depend on the decision variables x. If $f(x, \omega)$ is given explicitly, and $P(x, d\omega) = g(x, \omega) d\omega$ with $g(x, \omega)$ also given explicitly then it is possible to approximate $F(x)$ by the sample mean and to use conventional deterministic optimization techniques. Alternatively, one can use the SQG approach with the stochastic gradient

$$\xi^s = f_x(x^s, \omega^s) + f(x^s, \omega^s) \frac{g_x(x^s, \omega^s)}{g(x^s, \omega^s)}$$

where ω^s is an observation from $P(x^s, d\omega)$. In the case of implicitly given $P(x, d\omega)$, when ω is observed through Monte-Carlo simulation or on-line experiments, SQG methods provide the opportunity to solve the problem by the finite difference approach described earlier. Here is an analogue of the estimate (7) for (10):

$$\xi^s = \sum_{i=1}^{M} \frac{f(x^s + \Delta_s h^i, \omega^{s,i}) - f(x^s, \omega^{s,0})}{\Delta_s} h^i$$

when $P(x, d\omega) = P(d\omega)$. If $\omega^{s,0}, \omega^{s,1}, ..., \omega^{s,n}$ are independent observations from $P(d\omega)$, then $\mathrm{Var}\xi^s = O(\Delta_k^{-2})$. For smooth $f(x, \omega)$ it is possible to use the common random numbers $\omega^{k,0} = \omega^{k,1} = ... = \omega^{k,n}$ with variance that will not depend on Δ_s. Such a choice corresponds to a single run estimation of the gradient in combined simulation and optimization procedures.

Exploiting the problem structure. Stochastic decomposition: The efficiency of SQG methods can be considerably increased by exploiting the structure of specific problems. Let us consider, for example, a stochastic program with recourse for the case of discrete random parameters, which is equivalent to the following linear programming problem:

$$\min_x \left((c, x) + \sum_{k=1}^{K} p_k(d(k), y(k)) \right) \tag{11}$$

$$A(k)x + W(k)y(k) = b(k), \quad k = 1 : K$$

$$x \geq 0, \quad y(k) \geq 0$$

where p_k is the probability of the k-th random scenario. It is quite common to have the number of scenarios K very large and even infinite. For this case the stochastic quasigradient procedure provides a stochastic decomposition technique. For given x^s observe, at random, scenarios $k_1, ..., k_{\tau_s}$, $\tau_s \geq 1$ and solve the stochastic subproblem

$$\min \sum_{l=1}^{\tau_s} p_{k_l}(d(k_l), y(k_l))$$

$$W(k_l)y(k_l) = b(k_l) - A(k_l)x^s, \quad l = 1, ..., \tau_s$$

$$y(k_l) \geq 0, \quad l = 1, ..., \tau_s$$

Suppose that $u(k_l)$ are corresponding dual variables. Then the vector

$$\xi^s = c - \sum_{l=1}^{\tau_s} u^s(k_l)A(k_l)$$

is a stochastic quasigradient of the original objective function (11).

Nonstationary optimization. Learning and adaptation: Many applied problems can be formulated as optimization of an objective function in varying time under changing constraints. This refers to the case of on-line optimization through learning and adaptation, to deliberately designed optimization schemes with simultaneous approximation of "bad" objective and constraints functions by a sequence of "good" functions, and to dual approaches to the solution of constrained optimization problems where iterations are performed simultaneously in the space of primal and dual variables.

In this case an SQG method, on iteration s, performs one step of the minimization of function $F(s, x)$ using gradient $F_x(x, s)$ or the quasigradients described above. At the same time $F_x(x, s)$ can be considered as a quasigradient of the function $F(s+1, x)$, which defines the problem being minimized on the next iteration $s + 1$. Often such problems arise when the objective function (constraints) depends on unknown parameters, and the optimization takes place on-line with the estimation of these parameters. The unknown parameters may also correspond to strategies of different external "players" affecting the decision making process.

Distributed optimization. Game theory: Suppose that there are N players and

$$F(x) = \sum_{i=1}^{N} f^i(x_1, ..., x_N)$$

is the welfare function. Each player $i = 1 : N$ has only information about the welfare function $f^i(x_1, ..., x_i, ..., x_N)$ and may influence only his strategy x_i. If at the iteration (current moment) $s = 0, 1, ...$, each player attempts to improve his welfare in the direction $f^i_{x_i}(x^s_1, ..., x^s_i, ..., x^s_N)$ then the vector

$$(f^1_{x_1}(x^s), ..., f^N_{x_N}(x^s)),$$

under special assumptions, can also be considered as a quasigradient of the function $F(x)$.

Neural nets: This powerful technique emerged in pattern processing, classification and behavioral sciences. In fact, the training of a neural net consists of the minimization of the error function

$$F(x) = \sum_{i=1}^{N} F(i, x) \tag{12}$$

where each function $F(i, x)$ corresponds to one training example. The most frequently used algorithm for this purpose is called backpropagation and can be expressed as follows:

$$x^{s+1} = x^s - \rho_s F_x(m(s), x) \tag{13}$$

where the function $m(s)$ takes values from the set $\{1, ..., N\}$. It is possible to show that under quite general assumptions about $m(s)$ the vector $F_x(m(s), x)$ satisfies the quasigradient condition (4) or its generalization.

Optimization of Discrete Event Dynamic Systems (DEDS): This is an important class of problems that emerged recently in connection with mathematical modeling of modern production systems, telecommunication networks, and computer communications. Such systems are characterized by a state z^k, which evolves according to the state equation

$$z^{k+1} = \Theta(z^k, x, \omega^k) \tag{14}$$

where ω^k are independent random variables. The discrete event property implies that the mapping $\Theta(\cdot,\cdot,\cdot)$ depends discontinuously on its arguments. When the DEDS evolution consists of K steps, the DEDS optimization problem can be expressed as follows:

$$\min_{x\in X} \int f(\bar{z}^K, x, \bar{\omega}^K) \prod_{i=1}^{K} \varphi(x, z^i, \omega^i) \prod_{i=1}^{K} d\omega^i$$

where $\bar{z}^K = (z^1, ..., z^K)$, $\bar{\omega}^K = (\omega^1, ..., \omega^K)$, $\varphi(x, z^i, \omega^i)$ is the distribution function of ω^i. Here the function $f(\cdot,\cdot,\cdot)$ depends discontinuously on its arguments due to equation (14). This makes the problem difficult, because many gradient estimation schemes depend on the ability to change the order of differentiation and integration. Nevertheless, it is possible to develop various techniques for providing statistical estimates of the gradient that satisfy the quasigradient property (4). The development of such techniques is the subject of the sensitivity analysis of DEDS. Applied to DEDS, stochastic quasigradient methods become concurrent simulation and optimization algorithms.

5. SUMMARY

As we have seen, Stochastic Quasigradient Methods have applications to various difficult areas of optimization and operations research where the structure of the problem does not allow the use of highly developed deterministic software. In particular, their application can become one of the very few viable alternatives in cases of large-scale, highly nonlinear, multiextremal, nonsmooth, discontinuous, dynamic, stochastic optimization problems. (Such problems are not uncommon, and a challenge indeed!) They require comparatively modest computer resources per iteration, and are able to reach the vicinity of an optimal solution reasonably rapidly, which is sufficient for many applications.

****

BULLETIN BOARD

MPS-15: IMPORTANT DATES AND DEADLINES

The 15th International Symposium on Mathematical Programming will be held from August 15-19, 1994 on the campus of the University of Michigan, Ann Arbor, Michigan, USA. Some important dates and deadlines are as follows:

- 29 April 1994: Early registration deadline
- 1 May 1994: Last date to send paper copy of abstract
- 1 June 1994: Last date to send abstract by e-mail
- 15 July 1994: Last date for conference-booked hotel or Residence Hall reservations
- 18 July 1994: Deadline to request cancellation refund
- 22 July 1994: Fee for cancellation of Residence Hall lodging begins

An e-mail copy of the entire Second Announcement, which gives extensive details, can be obtained from: xvismp@um.cc.umich.edu

CONFERENCE ANNOUNCEMENT

MATHEMATICS OF NUMERICAL ANALYSIS REAL NUMBER ALGORITHMS

A Four-Week Conference starting July 16, 1995

Steve Smale (smale@math.berkeley.edu)

The mathematical theory of real number algorithms is to be the subject of this conference. Thus numerical analysis will be central with emphasis on geometrical, algebraic, analytic and foundational perspectives. Investigation of efficiency will play a special role. Practical algorithms will be the subject of theoretical analysis, but immediate useful results will not be demanded.

It is to be hoped that the conference will give the subject of numerical analysis a greater coherence through a focus on the mathematical side. In particular, an aim to strengthen the unity of mathematics and numerical analysis, and to narrow the gap between pure and applied mathematics. That goal is appropriate since many of the heroes of pure and applied mathematics, Newton, Euler, Lagrange and Gauss among them, established the basic real number algorithms. With the revolution of the

computer and the great achievements of scientific computation, it does service to both the pure and applied communities to support the mathematical development of numerical analysis.
This is an appropriate time to schedule such a meeting in view of the rapid development of heuristic work, a good base of theoretical work, and a widespread desire for mathematical deepening of the subject.
The starting date is July 16, 1995 and is to last 4 weeks. The place is to be Park City, Utah. Appropriate reservations have been made by the American Mathematical Society.

The conference will be international in character with strong representation from the most mathematically developed parts of numerical analysis. Besides tutorials and short courses, seminars in the following areas are contemplated :

(a) Linear Algebra

(b) Non-linear systems-path following

(c) Differential equations

(d) Linear programming problems

(e) Algebraic questions

(f) Foundations

(g) Information based complexity

(h) Lower bounds

(i) Approximation theory

Organizing Committee

Steve Smale, Chair, University of California, Berkeley
Gene Allgower, Colorado State University
Lenore Blum, MSRI, Berkeley
Alexandre Chorin, Berkeley
Philippe Ciarlet, Université Pierre et Marie Curie, Paris
Felipe Cucker, Universitat Pompeu Fabra, Barcelona
James Demmel, Berkeley
Ron DeVore, University of South Carolina
Gene Golub, Stanford
Arieh Iserles, University of Cambridge, England
Bert Jongen, RWTH Aachen, Germany
Herb Keller, Cal Tech
Jim Renegar, Cornell
Mike Shub, IBM, Yorktown Heights
Gil Strang, MIT
Henryk Woźniakowski, University of Warsaw and Columbia University

SELECTED UPCOMING ARTICLES FOR SIAM J. OPTIMIZATION

Convergence Theory of Nonlinear Newton–Krylov Algorithms *Peter N. Brown and Youcef Saad*
Line Search Procedures for the Logarithmic Barrier Function *Walter Murray and Margaret H. Wright*
On the Resolution of Linearly Constrained Convex Minimization Problems *Ana Friedlander, José Mario Martínez, and Sandra A. Santos*
Superlinearly Convergent $O(nL)$-Iteration Interior Point Algorithms for Linear Programming and the Monotone Linear Complementarity Problem *Kevin McShane*
On Optimization Problems with Variational Inequality Constraints *J. V. Outrata*
Triangular Decomposition Methods for Solving Reducible Nonlinear Systems of Equations *J. E. Dennis, Jr., José Mario Martínez, and Xiaodong Zhang*
Extension of Hoffman's Error Bound to Polynomial Systems *Xiao-Dong Luo and Zhi-Quan Luo*
Convergence Properties of a Class of Rank-two Updates *Paul T. Boggs and Jon W. Tolle*
Globally Convergent Inexact Newton Methods *Stanley C. Eisenstat and Homer F. Walker*
An Interior Point Column Generation Method for Linear Programming Using Shifted Barriers *John E. Mitchell*
Predictor-Corrector Methods for a Class of Linear Complementarity Problems *Sanjay Mehrotra and Robert A. Stubbs*
Can Parallel Branch and Bound Without Communication be Effective? *Per S. Laursen*
The Iterated Kalman Smoother as a Gauss–Newton Method *Bradley M. Bell*
Monotonicity of Primal and Dual Objective Values in Primal-Dual Interior-Point Algorithms *Shinji Mizuno, Michael J. Todd, and Levent Tuncel*
On Smoothing Exact Penalty Functions for Convex Constrained Optimization *Mustafa C. Pinar and Stavros A. Zenios*
The Monotonic Diameter of the Perfect 2-Matching Polytope *Fred J. Rispoli*
A Complexity Analysis for Interior-Point Algorithms Based on Karmarkar's Potential Function *Jun Ji and Yinyu Ye*
Problems of Hierarchical Optimization in Finite Dimensions *Ruoxin Zhang*
Stability Results for Stochastic Progrmas and Sensors, Allowing for Discontinuous Objective Functions *Zvi Artstein and Roger J.-B. Wets*
Local Convergence of a Two-Piece Update of a Projected Hessian Matrix *Chaya Gurwitz*
Partial Proximal Minimization Algorithms for Convex Programming *Dimitri P. Bertsekas and Paul Tseng*
Exposing Constraints *James V. Burke and Jorge J. Moré*

Potential Reduction Polynomial Time Method for Truss Topology Design *Aharon Ben-Tal and Arkadii Nemirovskii*

CONTRIBUTIONS TO THE V&N

The next issue (Fall, '94) will include essays by Mike Todd (Cornell; on complementarity) and Malcolm Pullan (Cambridge; on continuous LP).

Articles contributed by SIAG/OPT members are always welcome and can take one of two forms:

a) *Views*: short, scholarly, N^3 (Not Necessarily Noncontroversial) essay-type articles, say 2 to 4 pages long, on any topic in optimization and its interfaces with the sciences, engineering and education.

b) *News*: brief items for the Bulletin Board Section.

Our first preference is that a contribution take the form of a LaTeX file sent by email to the editor at the address given below. (If possible try it out in two-column format.) However, other forms of input are also acceptable.

The Bulletin-Board deadline for the next issue is September 15, 1994.

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

email: nazareth@wsumath.bitnet
or nazareth@amath.washington.edu