SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

No. 6 Spring 1995

NEW OPTIMIZATION METHODS ROOTED IN BIOLOGY AND PHYSICS

Traditional optimization methods are broadly applicable. However, during the past decade, considerable attention has been paid to new methods that are rooted in fundamental paradigms of biology and physics, and have evolved specifically in relation to these subjects.

Three primary examples are methods based on simulated annealing, genetic crossover and neural networks. They have significant overlap and exhibit some or all of the following characteristics that distinguish them from more traditional global optimization approaches:

- randomization techniques are employed in an essential way.
- optimization problems involving objective functions (potential surfaces) that are nonsmooth with numerous local minima are amenable to solution by virtue of their possessing an underlying statistical structure.
- parallelism arises inherently within the procedures.
- the requirement of monotonicity of objective functions values at successive iterates is *not* imposed.
- the methods are especially relevant for combinatorial (discrete) optimization, but can be formulated to handle continuous optimization problems as well.

A brilliant and eloquent book entitled *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, 1993, by Stuart Kaufmann (University of Pennsylvania and the Santa Fe Institute), in particular, Chapters 2 and 3, offers some fascinating insights into the relative merits of the above methods. Its broader content will also be of interest to the optimization community because, after all, evolution presents the ultimate example of a complex optimization problem.

Kaufmann introduces the NK model for describing selective evolution, for example, of RNA molecules or proteins for a well-defined fitness property, say the ability to catalyze a specific reaction. N refers to the number of parts in the system, for instance, gene loci in a genotype or amino acids in a protein. Each part (gene locus) has say 2 candidates (alleles), i.e., the total number of genotypes is 2^N, and mutation takes one genotype to another. (For example, each genotype has N one-allele mutant neighbours.) Each part makes a fitness contribution which depends on the allele at that part and the alleles at upto $K \leq N-1$ other parts among the N, i.e., K reflects the 'epistatic interactions' among the components. A method is proposed for assigning a 'fitness' to a genotype and the objective is to maximize fitness. As the main parameters are altered, the model generates a family of 'fitness landscapes' with a very interesting underlying statistical structure. This structure is studied in detail. The landscapes can vary from relatively smooth with highly correlated finess values, through increasingly rugged and multipeaked (perhaps in self-similar form), to maximally rugged with completely uncorrelated fitness values.

The NK model is closely related to an important class of models in statistical physics, called *spin-glasses*. A spin glass is characterized by particles with spin, either up or down, in mixture of attractive and repulsive interactions. A particular spin configuration has a total energy given by a Hamiltonian and the energy landscape it defines has many local minima owing to the mixture of interactions. (the latter description is taken from another important recent book, *The Computational Brain*, MIT Press, 1992, by P. Churchland and T. Sejnowski of UC-San Diego and the Salk Institute). The range of states or spin configurations explored by a spin-glass is a function of its temperature and is given, at equilibrium, by the 'Boltzmann distribution'. At high temperature the physical system is not trapped by local potential barriers (minima). At low temperature the system spends most of its time confined to deep potential wells.

Simulated annealing, developed by Kirkpatrick, Gelatt and Vecchi, originated in spin-glass physics and the 1953 Metropolis algorithm for simulating behaviour of a many-body system. It mimics a spin-glass by gradually lowering the analogue of temperature appropriate to the optimization model under consideration, whose cost function represents 'spin-glass energy'. In effect, the use of high 'temperature' smooths out the 'energy' landscape. As temperature is lowered, the landscape becomes progressively more rugged. If lowered too fast, the system like its physical counterpart can become trapped in a local minimum. However. if temperature is lowered slowly (the annealing schedule), a very good local energy minimum will usually be discovered. Kaufmann notes that annealing is a powerful strategy, but it only works well in landscapes in which deep energy wells also drain wide basins. It does not work well on either a random landscape or a 'golf course' potential (p.112).

Kaufmann further observes that there are profound similarities between the behaviour of a spin-glass physical system in a complex potential surface at a finite temperature and an adapting population on a rugged fitness landscape at a finite mutuation rate (p.43). The analogue of simulated annealing in population flow on rugged landscapes is to begin with a high mutation rate and gradually lower it. The search behaviour of a 'nearly melted' population may mimic important aspects of simulated annealing (p.112). However, he also says it is unlikely that the precise analogue of simulated annealing (tuning of mutation rate) plays a role in actual population search on a rugged fitness landscape.

Genetic recombination coupled with mutation is, of course, nature's search strategy of choice, and it provides the motivation for the genetic algorithm pioneered by Holland. The NK model also offers insight into the effectiveness of this algorithmic approach to optimization. In particular, Kaufmann observes that recombination is useless on uncorrelated landscapes, but can be effective under two conditions (1) when the high peaks are near one another and hence carry mutual information about their joint locations in genotype space and (2) when parts of the evolving system are quasi-independent of one another ($K << N$), and hence can be interchanged with modest chances that the recombined system has the advantages of both parents (p.95). He describes an interesting numerical experiment that illustrates this point (p.115).

Finally, it is interesting to note that spin-glasses also provided the analogies on which Hopfield was able to draw for his pioneering work on *neural networks* (spins represent neurons and their interactions represent synaptic strengths). The learning algorithms in neural network models are based on optimizing a cost function similar to a spin-glass Hamiltonian that can be visualized as a rugged fitness landscape. The two references mentioned above offer some interesting insights into this area as well. What is more, Kaufmann also discusses the promise of 'applied molecular evolution' (p.156-170), which has bearing on the recent breakthrough involving DNA computing with potentially major implications for optimization.

The foregoing paragraphs, quotations in the main, give only the merest glimpse into these important and currently esoteric optimization techniques. The two books and their extensive bibliographies will serve to draw the interested reader into the huge research literature on the subject.

Contents:

CHAIRMAN'S COLUMN

by Jorge J. Moré

The votes are in, and apparently Chicago politics is alive and well because the new chair has strong connections to Chicago. The new members of the board are

Jorge Moré, Chair
`more@mcs.anl.gov`

Tim Kelley, Vice Chair
`tim_kelley@ncsu.edu`

Ekkehard Sachs, Secretary/Treasurer
`sachs@uni-trier.de`

Philip Gill, Program Director
`pgill@ucsd.edu`

I have not given s-mail addresses because nowadays they are almost unnecessary. If needed, they can be found in the SIAG/OPT membership directory.

Before plunging ahead with our plans for the future, I want to thank the previous board for their work. As a member said, *This is a first rate organization, and I really appreciate the work of the officers.* Andy Conn served as chair; he also served as vice-chair for the previous two terms. Tim Kelley was the vice-chair, and we are fortunate that he will be serving this term also. He will provide much needed continuity and sanity. Jorge Nocedal served as secretary/treasurer. One of these days I will have to find out what Andy Conn meant by saying that "thanks to him we have no secrets and no money". David Gay, as program director, completed the board.

The first order of business is the SIAG/OPT prize. This prize is given to the best paper in optimization that has appeared in a peer-reviewed journal during the four years preceding the SIAM conference on optimization. As you know, the SIAM conference on optimization will be held May 20–22, 1996, in British Columbia, Canada, so we are looking for papers published in 1991–1995. Tim Kelley, the chair of the selection committee, will be issuing a call for nominations later on in this year, so start thinking of appropriate nominations. This requires some work on your part (a written justification and a citation not exceeding 25 words), but these awards bring recognition to researchers in our field.

What else does the SIAG/OPT do? Our purpose is to foster the development of theory, algorithms, and software for optimization, and the use of this technology in scientific and industrial applications. We organize the SIAM Optimization meeting and also minisymposia at other SIAM meetings.

I expect that the SIAG/OPT will sponsor and promote several workshops during the next three years. Ekkehard

Sachs has already expressed interested in organizing a workshop in Europe. We will keep you informed.

We also publish the Views and News newsletter. Larry Nazareth has been doing an outstanding job as editor, and his efforts have led to a classy publication. I always read it and find items of interest. We should be proud of this newsletter.

What else do we want to do? In my ballot statement I proposed several projects. Here is my original ballot statement:

The SIAG on Optimization should foster the development of optimization theory, algorithms, and software in areas of scientific and economic interest. We can do this by being aware of interesting optimization problems that arise in the applications areas and by making our work more accessible to other disciplines. The SIAM Conference on Optimization and the SIAG/OPT newsletter (Views and News) can be used to promote these issues. Sponsoring, organizing, and attending smaller, focused workshops is another way to address these issues. We could also set up an electronic newsletter (small, informal, not-too-frequent, moderated) for optimization problems in application areas, or a World Wide Web server on interesting optimization problems. A purpose of the server would be to collect stories (1–3 pages) on successful uses of optimization in applications.

Is there any interest in the server or the electronic newsletter?

The Web server would not be difficult to do, and it could be used to attract attention to optimization success stories. Research budgets are being decreased, and any information that would show how optimization technology (theory, algorithms, and software) benefits other areas would be helpful to program managers. Some of these success stories have appeared in SIAM News, but we could use more, and gathered in one place. If there is interest in this, we may even be able to get SIAM to do it.

The creation of another electronic newsletter or mailing list is more controversial. There are already several newsletters, newsgroups, and Web pages related to optimization. I receive and read the

`opt-net`

newsletter. This newsletter can also be accessed via the Web at

`http://moa.math.nat.tu-bs.de/opt-net`

The `na-net` newsletter and the associated Web page

`http://www.netlib.org/na-net`

is also of interest. The newsgroups

`sci.op-research`

`sci.math.num-analysis`

are another source of information. The Web page

`http://argus-inc.com/informs/informs.html`

has information on the INFORMS (Institute for Operations Research and Management Science), and

`http://silmaril.smeal.psu.edu/pol.html`

is the information service of INFORMS. In particular, this service has pointers to the linear programming and nonlinear programming FAQ (frequently asked questions) maintained by John Gregory.

I do not claim that the above provides a complete list of all the Web pages of general interest to optimizers. I would be interested in hearing of additional sources of information available on the Web.

Given all this material, is there a need for either the electronic newsletter or the Web page? Let me know what you think.

FORUM ESSAYS

OPTIMIZATION IN MACHINE LEARNING

by O. L. Mangasarian [1]

1. INTRODUCTION

Optimization has played a significant role in training neural networks [23]. This has resulted in a number of efficient algorithms [22,3,5,29,31] and practical applications in medical diagnosis and prognosis [34,35,27]. Other applications of neural networks abound [12,30,18,13] .

In this article, we focus on a number of problems of machine learning and pose them as optimization problems. Effective methods for solving these problems are also briefly described. For more details, the reader is referred to the extensive bibliography, in particular, [3,4,23]. Hopefully this work will point to further applications of optimization to the burgeoning field of machine learning.

2. MISCLASSIFICATION MINIMIZATION

A fundamental problem of machine learning is to construct (train) a classifier to distinguish between two or more disjoint point sets in an n-dimensional real space. A key factor in determining the classifier is the measure of

[1] Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, email: olvi@cs.wisc.edu. This material is based on research supported by Air Force Office of Scientific Research Grant F49620-94-1-000036 and National Science Foundation Grant CCR-9322479.

error used in constructing the classifier. We shall propose two error measures: one will merely count the number of misclassified points, while the other will measure the average distance of misclassified points from a separating plane. We will show that the first leads to an LPEC (linear program with equilibrium constraints) [24,20] while the second leads to a single linear program [21,4]. However, the problem of minimizing the number of misclassified points turns out to be NP-complete [11,17], but we shall indicate effective approaches [24,2] that render it more tractable.

For the sake of simplicity, we shall limit ourselves to discriminating between two sets, although optimization models apply readily to multicategory discrimination [6,7]. Let $\mathcal{A}$ and $\mathcal{B}$ be two disjoint point sets in R^n with cardinalities m and k respectively. Let the m points of $\mathcal{A}$ be represented by the $m \times p$ matrix A, while the k points of $\mathcal{B}$ be represented by the $k \times p$ matrix B. The integer p represents the dimensionality of the real space R^p into which the points of $\mathcal{A}$ and $\mathcal{B}$ are mapped by $F : R^n \longrightarrow R^p$, before their separation is attempted. In the simplest model $p = n$ and F is the identity map. However, more complex separation, say by quadratic surfaces [21], can be effected if one resorts to more general maps. (Note that complex separation, like fitting with high degree polynomials, is not always desirable, since it may lead to merely "memorizing" the training set.) The simplest and one of the most effective classifiers in R^p is the plane

$$xw = \theta \tag{1}$$

where $w \in R^p$ is the normal to the plane, $|\theta|/\|w\|_2$ is the distance of the plane to the origin in R^p, $x \in R^p$ is a point belonging to $F(\mathcal{A})$ or $F(\mathcal{B})$, and $\|\cdot\|_2$ denotes the 2-norm. The problem of training a linear classifier consists then of determining $(w, \theta) \in R^{p+1}$ so as to minimize the error criterion chosen. We note immediately that if the sets $F(\mathcal{A})$ and $F(\mathcal{B})$ are strictly linearly separable in R^p, then there exist $(w, \theta) \in R^{p+1}$ such that

$$\begin{aligned} Aw &\geqq e\theta + e \\ Bw &\leqq e\theta - e \end{aligned} \tag{2}$$

where e is a vector of ones of appropriate dimension. Since, in general (2) is not satisfiable, we attempt its approximate satisfaction by minimizing the chosen error criterion.

2.1 Number of Misclassified Points

Let $s : R \longrightarrow \{0, 1\}$ determine the step function that maps nonpositive numbers into $\{0\}$ and positive numbers into $\{1\}$. When applied to a vector $z \in R^p$, s returns a vector of zeros and ones in R^p, corresponding respectively to nonpositive and positive components z_i, $i = 1, \ldots p$, of z. The problem of minimizing the number of misclassified points then reduces to the following unconstrained minimization problem of a discontinuous function:

$$\begin{aligned} \min_{(w,\theta)\in R^{n+1}} &\|s(-Aw + e\theta + e)\| \\ &+\|s(Bw - e\theta + e)\| \end{aligned} \tag{3}$$

where $\|\cdot\|$ denotes some arbitrary, but fixed norm, on R^m or R^k. The sets $F(\mathcal{A})$ and $F(\mathcal{B})$ are linearly separable in R^p, if and only if the minimum of (3) is zero, and no points are misclassified, otherwise the minimum of (3) "counts" the number of misclassified points if the 1-norm is used. In [24] it was shown that (3) with the 1-norm is equivalent to the following LPEC:

$$\begin{array}{ll} \underset{w,\theta,r,u,s,v}{\text{minimize}} & er + es \\ & u + Aw - e\theta - e \geqq 0 \\ & r \geqq 0 \\ & r(u + Aw - e\theta - e) = 0 \\ & -r + e \geqq 0 \\ & u \geqq 0 \\ \text{subject to} & u(-r + e) = 0 \\ & v - Bw + e\theta - e \geqq 0 \\ & s \geqq 0 \\ & s(v - Bw + e\theta - e) = 0 \\ & -s + e \geqq 0 \\ & v \geqq 0 \\ & v(-s + e) = 0 \end{array} \tag{4}$$

It turns out that problem (4) is extremely difficult to solve. In fact, almost every point $(w, \theta) \in R^{p+1}$ is a stationary point, since a small perturbation of a plane $xw = \theta$ in R^p that does not contain points of either $F(\mathcal{A})$ or $F(\mathcal{B})$ will not change the number of misclassified points. In order to circumvent this difficulty, a parametric implicitly exact penalty function was proposed for solving (4) in [24] and implemented successfully in [2] by an approach that also identifies outlying misclassified points. A fast hybrid algorithm for approximately solving the misclassification minimization problem is also given in [11].

Another approach to solving (3) is by utilizing the highly effective smoothing technique [9,10] that has been used to solve many mathematical programs and related problems. In this approach, the step function $s(\zeta)$ is replaced by the classical sigmoid function of neural networks [18]:

$$s(\zeta) \cong \sigma(\zeta, \alpha) := \frac{1}{1 + e^{-\alpha\zeta}} \tag{5}$$

where α is a positive real number that approaches $+\infty$ for more accurate representation of the step function. With this approximation, the unconstrained discontinuous minimization problem is reduced to an unconstrained continuous optimization problem, that is however nonconvex. By letting α grow judiciously, effective computational schemes for tackling the NP-complete problem can be utilized. An important application of the misclassification error (3), is its use in constructing the more complex nonlinear neural network classifier of Section 3 below.

2.2 Average Distance of Misclassifications from Separating Plane

As early as 1964 [8,21], the distance of misclassified points from a separating plane was utilized to generate a linear programming problem for obtaining a separating plane (1) that approximately satisfied (2) by minimizing some measure of distance of misclassified points from the plane (1). Unfortunately, all these attempts [22,16,15] contained *ad hoc* ways for excluding the null solution ($w = 0$) that plagued a linear programming formulation for linearly inseparable sets. However, the robust model proposed in [4], which consists of minimizing the average of the 1-norm of the distances of misclassified points from the separating plane, completely overcame this difficulty. The linear program [4] proposed is this:

$$\begin{array}{ll} \underset{w,\theta,y,z}{\text{minimize}} & \frac{ey}{m} + \frac{ez}{k} \\ & Aw + y \geqq e\theta + e \\ \text{subject to} & Bw - z \leqq e\theta - e \\ & y, z \geqq 0 \end{array} \tag{6}$$

The key property of (6) is that it gives the null solution $w = 0$ if and only if $\frac{eA}{m} = \frac{eB}{k}$, in which case $w = 0$ is guaranteed to be not unique. Computationally, the LP (6) is very robust, rarely giving rise to the null solution, even in contrived examples where $\frac{eA}{m} = \frac{eB}{k}$. In the parlance of machine learning [18], the separating plane (1) is referred to as a "perceptron", "linear threshold unit" or simply "unit", with threshold θ and incoming arc weight w. This is in analogy to a human neuron which fires if the input $x \in R^p$, scalar-multiplied by the weight $w \in R^p$, exceeds the threshold θ.

3. NEURAL NETWORKS AS POLYHEDRAL REGIONS

A neural network can be defined as a generalization of a separating plane in R^p, and can be thought of as a nonlinear map: $R^p \rightarrow \{0, 1\}$. One intuitive way to generate such a map is to divide R^p into various polyhedral regions, each of which containing elements of $F(\mathcal{A})$ or $F(\mathcal{B})$ only. In its general form, this problem is again an extremely difficult and nonconvex problem. However, greedy sequential constructions of the planes determining the various polyhedral regions [22,25,1] have been quite successful in obtaining very effective algorithms for training neural networks much faster than the classical online (that is training on one point at a time) backpropagation (BP) gradient algorithm [32,18,26]. Online BP is often erroneously referred to as a descent algorithm, which it is not.

In this section of the paper we relate the polyhedral regions into which R^p is divided, to a neural network with one hidden layer of linear threshold units. It turns out that every such neural network can be related to a partitioning of R^p into polyhedral regions, but not the converse. However, any two disjoint point sets in R^p can be discriminated between by *some* polyhedral partition that corresponds to a neural network with one hidden layer with a sufficient number of hidden units [19,25].

We describe now precisely when a specific partition of R^p by h separating planes

$$xw^i = \theta^i, \ i = 1, \ldots, h, \tag{7}$$

corresponds to a neural network with h hidden units. The h separating planes (7) divide R^p into at most t polyhedral regions, where [14]

$$t := \sum_{i=0}^{p} \binom{h}{i}. \tag{8}$$

We shall assume that $F(\mathcal{A})$ and $F(\mathcal{B})$ are contained in the interiors of two mutually exclusive subsets of these regions. Each of these polyhedral regions can be mapped uniquely into a vertex of the unit cube in R^h,

$$\{z | z \in R^h, \ 0 \leqq z \leqq e\} \tag{9}$$

by using the map:

$$s(xw^i - \theta^i), \ i = 1, \ldots, h \tag{10}$$

where s is the step function defined earlier, and x is a point in R^p belonging to some polyhedral region. If the r polyhedral regions of R^p constructed by the h planes (7) are such that vertices of the cube (9) corresponding to points in $\mathcal{A}$, are linearly separable in R^h from the vertices of (9) corresponding to points in $\mathcal{B}$ by a plane

$$zv = \tau, \tag{11}$$

then the polyhedral partition of R^p corresponds to a neural network with h hidden linear threshold units (with thresholds θ^i, incoming arc weights w^i, $i = 1, \ldots, h$) and output linear threshold unit (with threshold τ and incoming arc weights v_i, $i = 1, \ldots, h$ [23]) . This condition is necessary and sufficient for the polyhedral partition of R^p in order for it to correspond to a neural network with one layer of hidden units. For more detail and graphical depiction of the neural network, see [23]. "Training" a neural network consists of determining $(w^i, \theta^i) \in R^{p+1}$, $i = 1, \ldots, h$, $(v, \tau) \in R^{h+1}$, such that the following nonlinear inequalities are satisfied as best as possible:

$$\begin{array}{l} \sum_{i=1}^{h} s(Aw^i - e\theta^i)v_i \geqq e\tau + e \\ \sum_{i=1}^{h} s(Bw^i - e\theta^i)v_i \leqq e\tau - e \end{array} \tag{12}$$

This can be achieved by minimizing the number of misclassified points in R^h by solving the following unconstrained minimization problem

$$\begin{array}{l} \min_{w^i,\theta^i,v,\tau} \|s(-\sum_{i=1}^{h} s(Aw^i - e\theta^i)v_i + e\tau + e)\| \\ +\|s(\sum_{i=1}^{h} s(Bw^i - e\theta^i)v_i - e\tau + e)\| \end{array} \tag{13}$$

where the norm is some arbitrary norm. If the square of the 2-norm is used in (13) instead of the 1-norm, and if the step function s is replaced by the sigmoid function in (13), we obtain an error function similar to the error function that BP attempts to find a stationary point for, and for which a convergence proof is given in [26], and stability analysis in [33]. We note that the classical exclusive-or (XOR) example [28] for which F is the identity map and $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$, gives a zero minimum for (13) with the following solution:

$$\begin{array}{l} (w^1, \theta^1) = ((2 \;\; -2),\ 1),\ (w^2, \theta^2) = ((-2 \;\; 2),\ 1) \\ (v, \tau) = ((2 \;\; 2),\ 1) \end{array} \tag{14}$$

It is interesting to note that the same solution for the XOR example is given by the greedy multisurface method tree (MSMT) [1]. MSMT attempts to separate as many points of $\mathcal{A}$ and $\mathcal{B}$ as possible by a first plane obtained by solving (6), and then repeats the process for each of the ensuing halfspaces, until adequate separation is obtained. For this example, the first plane obtained [4] is $(w^1, \theta^1) = ((2 \;\; -2),\ 1)$, which separates $\{(1,0)\}$ from $\{(0,0),(0,1),(1,1)\}$. The second plane obtained is $(w^2, \theta^2) = ((-2 \;\; 2),\ 1)$, separates $\{(0,1)\}$ from $\{(0,0),(1,1)\}$, and the separation is complete between $\mathcal{A}$ and $\mathcal{B}$. These planes correspond to a neural network that gives a zero minimum to (13), which of course is not always the case. However, MSMT frequently gives better solutions than those generated by BP and is much faster than BP.

References

[1] K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

[2] K. P. Bennett and E.J. Bredensteiner. A parametric optimization method for machine learning. Department of Mathematical Sciences Report No. 217, Rensselaer Polytechnic Institute, Troy, NY 12180, 1994.

[3] K. P. Bennett and O. L. Mangasarian. Neural network training via linear programming. In P. M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 56–67, Amsterdam, 1992. North Holland.

[4] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[5] K. P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization & Applications*, 2:207–227, 1993.

[6] K. P. Bennett and O. L. Mangasarian. Multicategory separation via linear programming. *Optimization Methods and Software*, 3:27–39, 1993.

[7] K. P. Bennett and O. L. Mangasarian. Serial and parallel multicategory discrimination. *SIAM Journal on Optimization*, 4(4):722–734, 1994.

[8] A. Charnes. Some fundamental theorems of perceptron theory and their geometry. In J. T. Lou and R. H. Wilcox, editors, *Computer and Information Sciences*, pages 67–74, Washington, D.C., 1964. Spartan Books.

[9] Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. Technical Report 1191, CS Department, U. of Wisconsin, Madison, November 1993. Mathematical Programming, to appear. Available from ftp://ftp.cs.wisc.edu/tech-reports/reports/93/tr1191.ps.Z.

[10] Chunhui Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. Technical Report 94-11, CS Department, U. of Wisconsin, Madison, August 1994. Computational Optimization and Applications, to appear. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/94-11.ps.Z.

[11] Chunhui Chen and O. L. Mangasarian. Hybrid misclassification minimization. Technical Report 95-05, CS Department, U. of Wisconsin, Madison, February 1995. Advances in Computational Mathematics, submitted. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/95-05.ps.Z.

[12] L. DeSilets, B. Golden, Q. Wang, and R. Kumar. Predicting salinity in the Chesapeake Bay using backpropagation. *Computers & Operations Research*, 19:277–285, 1992.

[13] S.I. Gallant. *Neural Network Learning and Expert Systems.* MIT Press, Cambridge, Massachusetts, 1993.

[14] G.M. Georgiou. Comments on hidden nodes in neural nets. *IEEE Transactions on Circuits and Systems*, 38:1410, 1991.

[15] F. Glover. Improved linear programming models for discriminant analysis. *Decision Sciences*, 21:771–785, 1990.

[16] R.C. Grinold. Mathematical methods for pattern classification. *Management Science*, 19:272–289, 1972.

[17] David Heath. *A geometric Framework for Machine Learning*. PhD thesis, Department of Computer Science, Johns Hopkins University–Baltimore, Maryland, 1992.

[18] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, California, 1991.

[19] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[20] Z.-Q. Luo, J.-S. Pang, D. Ralph, and S.-Q. Wu. Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. Technical Report 275, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Hamilton, Ontario L8S 4K1, Canada, 1993. Mathematical Programming, to appear.

[21] O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations Research*, 13:444–452, 1965.

[22] O. L. Mangasarian. Multi-surface method of pattern separation. *IEEE Transactions on Information Theory*, IT-14:801–807, 1968.

[23] O. L. Mangasarian. Mathematical programming in neural networks. *ORSA Journal on Computing*, 5(4):349–360, 1993.

[24] O. L. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5:309–323, 1994.

[25] O. L. Mangasarian, R. Setiono, and W. H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. In T. F. Coleman and Y. Li, editors, *Large-Scale Numerical Optimization*, pages 22–31, Philadelphia, Pennsylvania, 1990. SIAM.

[26] O. L. Mangasarian and M.V. Solodov. Serial and parallel backpropagation convergence via nonmonotone perturbed minimization. *Optimization Methods and Software*, 4(2):103–116, 1994.

[27] O. L. Mangasarian, W. Nick Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Technical Report 94-10, CS Department, U. of Wisconsin, Madison, 1994. Operations Research, to appear. Available from ftp://ftp.cs.wisc.edu/math-prog/tech-reports/94-10.ps.Z.

[28] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Massachusetts, 1969.

[29] S. Mukhopadhyay, A. Roy, and S. Govil. A polynomial time algorithm for generating neural networks for pattern classification: Its stability properties and some test results. *Neural Computation*, 5:317–330, 1993.

[30] K.E. Nygard, P. Juell, and N. Kadaba. Neural networks for selecting vehicle routing heuristics. *ORSA Journal on Computing*, 4:353–364, 1990.

[31] A. Roy, L.S. Kim, and S. Mukhopadhyay. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks*, 6:535–545, 1993.

[32] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, pages 318–362, Cambridge, Massachusetts, 1986. MIT Press.

[33] M.V. Solodov and S.K. Zavriev. Stability properties of the gradient projection method with applications to the backpropagation algorithm. CS Department, Math. Prog. Tech. Report 94-05, U. of Wisconsin, Madison, June 1994. SIAM Journal on Optimization, submitted.

[34] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences, U.S.A.*, 87:9193–9196, 1990.

[35] W. H. Wolberg, W. N. Street, and O. L. Mangasarian. Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters*, 77:163–171, 1994.

****

PATTERN SEARCH METHODS FOR NONLINEAR OPTIMIZATION

by Virginia Torczon [1]

1. INTRODUCTION

In 1961, Hooke and Jeeves [8] coined the term *direct search*

> ...to describe sequential examination of trial solutions involving comparison of each trial solution with the "best" obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be.

[1] Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892. e-mail: va@rice.edu

In the intervening years, the term *direct search* has come to refer to any method that does not use derivatives or approximations of derivatives to solve the problem

$$\min_x f(x),$$

where $x \in \mathbf{R}^n$ and $f : \mathbf{R}^n \to \mathbf{R}$. Instead the search is directed using only function values. Thus direct search properly includes such disparate methods as the pattern search method proposed by Hooke and Jeeves [8], the wildly popular genetic algorithms [7], the Nelder-Mead simplex method [11] (a Science Citation Classic), and random search methods [2].

Few of the direct search methods were conceived with any sort of convergence analysis in mind. Notable exceptions include the conjugate directions method proposed by Powell [15,25] and the class of deformed configuration methods proposed by Rykov and his colleagues [18,17,19,9].

However, recent work [23] shows that a significant subset of the direct search algorithms—a class we call *pattern search* methods in deference to Hooke and Jeeves—share a structure that makes a unified convergence analysis possible for the algorithms as originally conceived. The surprising conclusion of this analysis is that global convergence results comparable to those for line search [12] and trust region [10] globalization strategies are possible, even though gradient information is neither explicitly calculated nor approximated.

The key to the convergence analysis is that we are able to relax the conditions on accepting a step by placing stronger conditions on the step itself.

2. GENERALIZED PATTERN SEARCH

Pattern search methods follow the general form of most optimization methods: given an initial guess at a solution x_0 and an initial choice of a step length parameter $\Delta_0 > 0$,

```
ALGORITHM 1. General Pattern Search
For $k = 0, 1, \ldots,$
  a) Check for convergence.
  b) Compute $f(x_k)$.
  c) Determine a step $s_k$ using Exploratory Moves($\Delta_k, P_k$).
  d) If $f(x_k) > f(x_k + s_k)$, then $x_{k+1} = x_k + s_k$. Otherwise $x_{k+1} = x_k$.
  e) Update($\Delta_k, P_k$).
```

Pattern search methods only require *simple*, as opposed to *sufficient*, decrease on the objective function. This weaker condition is possible because we require that the step be defined by Δ_k and the *pattern* P_k and we place certain mild conditions on both the Exploratory Moves and the way in which we update Δ_k to guarantee global convergence. We do not need any derivative information because we do not need to enforce the classical sufficient decrease conditions, such as the Armijo–Goldstein–Wolfe conditions used for line search methods or the fraction of Cauchy decrease or fraction of optimal decrease conditions used for trust region methods.

A pattern P_k is defined by two components, a real nonsingular *basis matrix* $B \in \mathbf{R}^{n\times n}$ and an integer *generating matrix* $C_k \in \mathbf{Z}^{n\times p}$, where $p > 2n$. In addition, the columns of C_k must contain a *core pattern* represented by $M_k \in \mathbf{M} \subset \mathbf{Z}^{n\times n}$ and its negative $-M_k$, where $\mathbf{M}$ is a finite set of nonsingular matrices (thus ensuring that C_k has full row rank).

A pattern P_k is then defined by the columns of the matrix $P_k = BC_k$. Since both B and C_k have rank n, the columns of P_k span $\mathbf{R}^n$. The steps are of the form $s_k = \Delta_k B c_k$, where $c_k \in C_k$. (We adopt this convenient abuse of notation to indicate that c_k is a column of C_k.)

We require that the Exploratory Moves satisfy two hypotheses:

HYPOTHESES ON EXPLORATORY MOVES.

1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k$.
2. If $\min\{f(x_k + y),\ y \in \Delta_k B[M_k, -M_k]\} < f(x_k)$, then $f(x_k + s_k) < f(x_k)$.

The second hypothesis is more interesting. It suggests that if descent can be found for any one of the $2n$ steps defined by the core pattern, then the Exploratory Moves must return a step that gives simple decrease. There is no requirement that such a step must be defined by the core pattern, nor that all $2n$ steps defined by the core pattern must be evaluated, or even that the step returned give the greatest decrease possible.

Thus, a legitimate Exploratory Moves algorithm would be one that somehow "guesses" which of the steps defined by $\Delta_k P_k$ will produce simple decrease and then evaluates only that single step. At the other extreme, a legitimate Exploratory Moves algorithm would be one that evaluates all p steps defined by $\Delta_k P_k$ and returns the step that produced the least function value.

The core pattern guarantees that at least one of the $2n$ directions defined by the columns of $[M_k, -M_k]$ is a descent direction when $\nabla f(x_k) \neq 0$. Thus the Exploratory Moves algorithm must contain a safeguard to ensure that these $2n$ directions are polled if the other strategies employed do not produce a step that gives simple decrease on $f(x_k)$.

To finish our specification we give a simplification of the technique for updating the step length control parameter Δ_k. (For a full specification of the update procedures for both Δ_k and $P_k = BC_k$, see [23].)

ALGORITHM 2. (Simplified) Update for Δ_k.
Given $\tau = 2$, $\theta = \tau^{-1}$ and $\lambda_k \in \{\tau^0, \tau^1\}$.

a) If $f(x_k + s_k) < f(x_k)$, then $\Delta_{k+1} = \lambda_k \Delta_k$.

b) Otherwise, $\Delta_{k+1} = \theta \Delta_k$.

Here Δ_k may be reduced if and only if simple decrease has not been realized.

The general specification for pattern search methods is rich enough to capture a variety of direct search algorithms. These include coordinate search with fixed step lengths (Davidon [4] describes its use by Fermi and Metropolis to set phase shift parameters), the evolutionary operations algorithm of G.E.P. Box [1], the pattern search method of Hooke and Jeeves [8], and the multidirectional search algorithm of Dennis and Torczon [6,21].

The general specification also leads to global convergence results. The goal of the next section is to show that pattern search methods are as robust as their proponents have long claimed and to demonstrate that the convergence analysis is comparable to that for line search and global trust region strategies.

3. ANALYSIS

Critical to proving global convergence of pattern search methods is recognizing the algebraic structure of the iterates, a structure that is independent of the function to be optimized. This leads to the following theorem.

Theorem 1: Any iterate x_N produced by a general pattern search (Algorithm 1) can be expressed in the following form:

$$x_N = x_0 + \left(\beta^{r_{LB}} \alpha^{-r_{UB}}\right) \Delta_0 B \sum_{k=0}^{N-1} z_k,$$

where

- $\beta/\alpha \equiv \tau$, with $\alpha, \beta \in \mathbf{N}$ and relatively prime, and τ is as defined in the algorithm for updating Δ_k (Algorithm 2),
- r_{LB} and r_{UB} *depend on* N,
- $z_k \in \mathbf{Z}^n$, $k = 0, \ldots, N-1$.

The import of this theorem is that all the iterates lie on a scaled, translated integer lattice. The basis depends on the initial choice of Δ_0 and the basis matrix B, the translation depends on the initial choice of x_0, and the scaling is based solely on the sequence of updates that have been applied to Δ_k, for $k = 0, \ldots, N-1$. (See [23] for a proof.)

With this theorem in hand, it is then possible to prove the following theorem regarding the global convergence behavior of pattern search methods.

Theorem 2: Assume that $L(x_0) = \{x : f(x) \leq f(x_0)\}$ is compact and that $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is continuously differentiable on $L(x_0)$. Then for the sequence of iterates $\{x_k\}$ produced by the general pattern search (Algorithm 1),

$$\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0.$$

There are three key points to the proof [23]. First, it is straightforward to show that pattern search methods are descent methods. Second, it is possible to prove that pattern search methods are gradient-related methods (as defined in [13]). The third and final part of the argument involves a proof by contradiction to show that the algorithm cannot terminate prematurely due to inadequate step length control mechanisms.

The proof that pattern search methods are descent methods uses differentiability, the core pattern, the Hypotheses on the Exploratory Moves and the update rules for Δ_k. The n columns of BM_k form a set that spans $\mathbf{R}^n$ so that if $\nabla f(x_k) \neq 0$, then at least one of the $2n$ directions defined by the columns of $B[M_k, -M_k]$ (the core pattern) must be a descent direction from the current iterate. The Hypotheses on the Exploratory Moves require that in the worst case, if we have not already found a step that produces simple decrease, then we must look at all $2n$ steps defined by $\Delta_k B[M_k, -M_k]$. The update for Δ_k specifies that Δ_k must be reduced if the Exploratory Moves failed to produce a step giving simple decrease. Thus we have a backtracking line search along $2n$ search directions, at least one of which is a direction of descent. It is then a simple matter to show that this process must terminate in a finite number of iterations.

To show that pattern search methods are gradient-related methods, we prove that the following holds for any $x \neq 0$:

$$\max \left\{ \frac{|x^T s_k^i|}{\|x\| \|s_k^i\|},\ i = 1, \ldots, p \right\} \geq \xi > 0,$$

with

$$\xi = \min_{M \in \mathbf{M}} \left\{ \frac{1}{\kappa(BM)\sqrt{n}} \right\},$$

where $\kappa(BM)$ denotes the condition number of the matrix BM. (All norms are the Euclidean vector norms.) Again we make use of the core pattern defined by $B[M_k, -M_k]$, of the fact that $M_k \in \mathbf{M}$ where $\mathbf{M}$ is a finite set, and of the fact that the Hypotheses on Exploratory Moves require that all steps satisfy $x_k \in \Delta_k P_k$.

The most delicate part of the analysis involves assuring that the common pathologies that require step length control mechanisms cannot occur because of the structure placed on the choice of iterates for pattern search methods. The problem of steps that are either too long, relative to the amount of decrease realized by the next iterate, or too short, relative to the amount of decrease predicted by the gradient at the current iterate [5], cannot occur. Because the iterates lie on a lattice, which depends on Δ_k, steps of arbitrary lengths along arbitrary search directions are not possible. Thus such pathologies cannot occur. Details of the proof can be found in [23].

Proofs of global convergence for individual pattern search methods have appeared in the literature over the years. The text by Céa [3] contains a proof of convergence for the pattern search method of Hooke and Jeeves while the text by Polak [14] contains a proof of convergence for coordinate search with fixed step length. These two proofs require that the step sizes be monotonically decreasing. Yu Wen-ci [24] gives a unified analysis for a class of direct search techniques, but it requires both that the step sizes be monotonically decreasing and that an "error-controlling" sequence, which plays the role of a sufficient decrease condition, be introduced into the algorithms considered—with no suggestions on how such a sequence could be constructed in practice. As we have demonstrated, neither restriction is necessary to obtain global convergence results for pattern search methods.

References

[1] George E. P. Box, Evolutionary Operation: A Method for Increasing Industrial Productivity, *Applied Statistics*, 6, 81–10, 1957.

[2] M. J. Box, D. Davies and W. H. Swann, *Non-Linear Optimization Techniques*, ICI Monograph No. 5, Oliver & Boyd, Edinburgh, 1969.

[3] Jean Céa, *Optimisation : théorie et algorithmes*, Dunod, Paris, 1971.

[4] William C. Davidon, A belated preface for ANL 5990, *SIAM J. Optimization*, 1, 1-1, 1991.

[5] Dennis, Jr., J. E. and Robert B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[6] Dennis, Jr., J. E. and Virginia Torczon, Direct Search Methods on Parallel Machines, *SIAM J. Optimization*, 1, 448–474, 1991.

[7] John H. Holland, Genetic Algorithms, *Scientific American*, 66–72, July, 1992.

[8] Robert Hooke and T. A. Jeeves, 'Direct Search' Solution of Numerical and Statistical Problems, *J. Assoc. Comput. Mach.*, 8, 212–229, 1961.

[9] Alexander G. Kuznetsov, Nonlinear Optimization Toolbox, Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K., OUEL 1936/92, 1992.

[10] J. J. Moré, Recent Developments in Algorithms and Software for Trust Region Methods, in *Mathematical Programming, The State of the Art*, A. Bachem and M. Grotschel and G. Korte (Eds.), Springer–Verlag, 256–287, 1983.

[11] J. A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.*, 7, 308–313, 1965.

[12] Jorge Nocedal, Theory of Algorithms for Unconstrained Optimization, *Acta Numerica*, 1, 199-242, 1992.

[13] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[14] E. Polak, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.

[15] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Comput. J.*, 7, 155–162, 1994.

[16] H. H. Rosenbrock, An Automatic Method for finding the Greatest or Least Value of a Function, *Comput. J.*, 3, 175–184, 1960.

[17] A. S. Rykov, Simplex Methods of Direct Search, *Engineering Cybernetics*, 18, 12–18, 1980.

[18] A. S. Rykov, Simplex Direct Search Algorithms, *Automation and Remote Control*, 41, 784–793, 1980.

[19] A. S. Rykov, Simplex Algorithms for Unconstrained Minimization, *Problems of Control and Information Theory*, 12, 195–208, 1983.

[20] W. Spendley and G. R. Hext and F. R. Himsworth, Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation, *Technometrics*, 4, 441–461, 1962.

[21] Virginia Torczon, On the Convergence of the Multidirectional Search Algorithm, *SIAM J. Optimization*, 1, 123–145, 1991.

[22] Virginia Torczon, PDS: Direct Search Methods for Unconstrained Optimization on Either Sequential or Parallel Machines, Department of Mathematical Sciences, Rice University, Houston, TX 77251–1892, No. 92–9, 1992.

[23] Virginia Torczon, On the Convergence of Pattern Search Algorithms, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251–1892, No. 93–10, 1993 (In revision; submitted to *SIAM Journal on Optimization*).

[24] Yu Wen-ci, Positive Basis and a Class of Direct Search Techniques, *Scientia Sinica*, Special Issue of Mathematics, 1, 53–67, 1979.

[25] Willard I. Zangwill, Minimizing a function without calculating derivatives, *Comput. J.*, 10, 293–296, 1967.

****

If QUASI-NEWTON then WHY NOT QUASI-CAUCHY? endif

by J.L. Nazareth [1]

1. INTRODUCTION

When Davidon's variable-metric algorithm [4] was introduced to the optimization community in the early nineteen-sixties by Fletcher and Powell [8] its performance vis-a-vis other methods then popular for minimizing a smooth nonlinear function, say $f(x)$, $x \in R^n$, is described by Powell [13] in a recent historical account as follows: "I tried the method on some examples and the results were stunning".

A significant advantage of the variable-metric (more accurately, metric-based quasi-Newton) algorithm derives from its update of an approximation to the Hessian matrix of second derivatives using gradient vectors at successive widely separated iterates. Thus it only requires first order derivative information. Its proven effectiveness tilted the balance away from Newton's method, which requires potentially expensive Hessian evaluations or estimates obtained by differences; Cauchy's method (steepest descent possibly with diagonal scaling), which uses only first order derivatives and is robust, but usually quite slow; and pattern (and cyclic) search methods, many with P.G. Woodhousian names (Hooke and Jeeves, Nelder and Mead, Rosenbrock), which are derivative-free, require much weaker assumptions on f, are elegant and simple to implement, but usually only work well when n is small.

Today the balance is tilting back. Newton's method is enjoying a resurgence fuelled by automatic differentiation, see Griewank [10]. Parallel computing, along with the need to find global minima of functions with a complex topography (for example, noisy functions), has resurrected pattern search, see Dennis and Torczon [5]. And, of course, Cauchy's method has always remained a useful way to guarantee convergence and is thus a fallback measure.

In the foregoing repertoire of methods, there is an obvious progression. Quasi-Newton algorithms occupy the middle ground between Newton's method and Cauchy's method. Beyond (or beneath) are derivative-free methods, and a natural question arises: Is there also a middle ground between Cauchy's method and derivative-free pattern search? In full generality this question has indeed already been addressed, most notably in the stochastic quasi-gradient algorithms pioneered by Ermoliev (see Ermoliev and Gaivoronski [7] for an overview) with antecedents in the methods of Shor [14] and others. However, these are inherently *non-monotonic* algorithms designed especially for non-smooth functions and they employ a step-length strategy quite different from the line search used in the usual metric-based gradient-related descent setting. (Another interesting inherently nonmonotonic Cauchy algorithm when f is smooth is that of Barzilai and Borwein [1].)

In contrast, our concern is with inherently monotonic gradient-related algorithms for minimizing differentiable functions with the added feature that the gradient vector at any iterate, say x, is not available directly, or inexpensively via automatic differentiation, and for which the option of taking n finite-differences of function values at x (over numerically infinitesmal steps along, for example, the coordinate axes) is prohibitively expensive.

Let us agree that by a metric-based quasi-Cauchy algorithm for minimizing functions of the foregoing type we mean an algorithm that

- develops derivative-related search directions that are inherently directions of *descent*,
- employs a *line search* that does not require gradient vectors,
- and utilizes a metric defined by *diagonal* matrices.

We discuss very simple and specific techniques for each of these three items in the next section. A computational experiment is described in Section 3. Finally the much broader ramifications of these ideas are outlined in Section 4.

2. A QC ALGORITHM

Let D be a diagonal matrix with positive diagonal elements. At any point, say x with gradient vector $g = \nabla f(x)$, the Cauchy method uses the direction of steepest descent in the metric defined by D, namely, $-D^{-1}g$.

Now, form instead an approximation $\bar{g}$ (termed the quasi-gradient) to g in a "natural" way that ensures that the direction $d = -D^{-1}\bar{g}$ is a direction of descent. Conduct a line search along d, update D to a diagonal matrix D_+ so as to incorporate curvature information gathered from the line search, and then repeat the process.

2.1 Descent Direction: Within the foregoing context, the technique for defining search direction at x will be based a simple lemma as follows:

[1] Department of Pure and Applied Mathematics, Washington State University, Pullman, WA 99164.
e-mail: nazareth@amath.washington.edu.

Lemma: Let $E = \{e_1, \ldots, e_k\}$ be a set of non-zero directions at the point x with gradient vector g, and let D be a positive definite diagonal matrix. Let $\bar{g}$ be any non-zero vector that approximates g in the sense that

$$\bar{g}^T e_i = g^T e_i, \quad 1 \leq i \leq k. \tag{1}$$

If $d \equiv -D^{-1}\bar{g}$ is linearly dependent on $e_1, \ldots, e_k$ then d is a direction of descent.
Proof: From (1), $(\bar{g} - g)^T e_i = 0$, $i = 1, \ldots, k$. If d is linearly dependent on $e_i, \ldots, e_k$ then $(\bar{g} - g)^T d = 0$. Thus $g^T d = -\bar{g}^T D^{-1}\bar{g} < 0$. □.

Take $2 \leq k << n$, and group the n coordinate vectors into mutually exclusive blocks each containing $(k-1)$ vectors, These say K blocks are used one at a time in in a cyclic sequence. At the current iterate, say x, the directional derivative along each vector in the the block that comes up next in the sequence (the 'current block') is estimated by finite differences in the standard way using, for example, an $O(macheps^{1/2})$ step from x along the vector. In addition, as we will see in the next subsection 2.2, the line search along the previous search direction, say d_-, that found the current iterate x, will exit with the directional derivative at x along d_-, i.e., the quantity $g^T d_-$. Thus, the set of k vectors consisting of the $(k-1)$ vectors in the current block and the vector d_- will constitute the vectors $e_1, \ldots, e_k$ of the lemma. These vectors have known directional derivatives at x.

The quantity $\bar{g}$, termed the quasi-gradient, is then found by solving the low-dimensional linear least squares problem: minimize the euclidean norm of $\bar{g}$ subject to satisfying the constraints (1). This can be done efficiently (see also subsection 4.1).

Finally, the search direction at x is defined by $d = -D^{-1}\bar{g}$. The choice of the matrix D defining the metric is considered in subsection 2.3. The directional derivative $g^T d$ along d is obtained by a finite difference. If this establishes that d is a direction of descent then proceed to the line search. If not, then the lemma tells us that d must be linearly independent of the set E. Add d and and its directional derivative to the set of directions in the lemma that are used to obtain the quasi-gradient i.e., $k \leftarrow k+1$, $E \leftarrow E \cup d$, and repeat the procedure to obtain a new quasi-gradient $\bar{g}$, search direction $d = -D^{-1}\bar{g}$ and directional derivative. Obviously it must terminate in a finite number of repetitions with a direction of descent.

Refinements of this basic approach that rely more heavily on the foregoing lemma are outlined in subsection 4.1.

2.2 Line Search: The direction d and its directional derivative $g^T d$ are used to initiate a standard line search based, for example, on fitting a cubic polynomial to function and directional derivative information. When the line search defines a fresh point, say x_+ along the line passing through x parallel to d and requires the directional derivative, say $g_+^T d$ at x_+, this information is *obtained again by a finite difference operation.* The line search is terminated when the Wolfe conditions are satisfied or, more simply, when $|g_+^T d|/|g^T d| \leq acc$ where $acc \in (0,1)$. These conditions imply $(g_+^T d - g^T d) > 0$.

2.3 Diagonal Metric: The metric used to define the search direction d at x will be defined by a diagonal matrix, say D, with positive diagonal elements. The line search along $d = -D^{-1}\bar{g}$ is entered at x with slope information $g^T d$ and exits at x_+ with slope information $g_+^T d$. Define $s = x_+ - x \neq 0$, $y = g_+ - g$ (the vector y is *not* available to the algorithm), $a = s^T D s > 0$ and $b = y^T s$ (available by computing $b = (g_+^T d - g^T d)(||s||/||d||) > 0$). Note that $b > 0$.

The new curvature information derived from the line search is incorporated into the updated diagonal matrix, say D_+, that defines the metric at x_+ in the following way: Let M_+ be the matrix obtained by updating D by a matrix of rank one, namely,

$$M_+ = D + \frac{(b-a)}{a^2} D s s^T D, \tag{2}$$

and let D_+ be the diagonal matrix with $(D_+)_{ii} = (M_+)_{ii}, i = 1, \ldots, n$. This matrix can be computed directly as follows:

$$(D_+)_{ii} = D_{ii} + \frac{(b-a)}{a^2} s_i^2 D_{ii}^2.$$

Note that directional derivative information from the computation of the search direction and from the line search is being *fully* used, and that only 'hard' information (as constrasted with 'soft' quasi-gradient estimates) is used to update the metric.

The matrix M_+ satisfies the weak quasi-Newton (secant) relation of Dennis and Wolkowicz [6]

$$s^T M_+ s = b \equiv y^T s.$$

Compare with the usual quasi-Newton relation $M_+ s = y$. Under the foregoing conditions $a > 0$, $b > 0$, it is easy to show that M_+ defined by (2) is positive definite (see [6]). Thus D_+ has positive diagonal elements. There is a subtle difference in our usage here, because Dennis and Wolkowicz [6] assume that the vector y is available (their setting is quasi-Newton) and much of their theoretical development involves updates defined in terms of y or the quantity $c = y^T D^{-1} y$. An exception is the update labelled (4.10) in [6], which corresponds to (2) above provided b is computed from directional derivatives instead of gradients.

Note that the foregoing matrix D_+ does *not* in general satisfy the weak secant relation. Much more can be said about diagonal updating in a quasi-Cauchy setting as outlined in subsection 4.2.

3. COMPUTATIONAL EXPERIMENT

The foregoing algorithm was encoded in Matlab without any frills or fine-tuning, and run with the setting $acc = 0.5$ in the line search, the choice $(k-1) = 5$ in the QC method, and again with the choice $(k-1) = n$ that in effect yielded the Cauchy method (with the diagonal scaling of subsection 2.3). In addition, the cyclic coordinate descent method with a line search was implemented in Matlab for purposes of comparison. The three algorithms were run on a trignometric function, which is a standard test problem given in More et al [11] using the standard starting point and $n = 30$. (The optimal function value is 0.)

The following table gives function evaluation counts and corresponding function values for the three algorithms, in each case recorded when the count first exceeded 500 and thereafter in approximate multiples of 100 until 1000 was exceeded.

Quasi-Cauchy		Cauchy		Cyclic Coord.	
507	5.62	502	190.80	511	27.77
600	3.89	600	121.26	601	22.67
701	0.051	698	92.04	727	0.49
806	6.84e-4	796	60.23	817	0.35
900	4.81e-5	896	39.26	908	0.34
1003	8.71e-7	1026	29.23	1020	0.013

The Matlab program can be sent by e-mail to anyone interested in more detail, and will also serve to provide more complete documentation of the experiment.

Of course "one swallow does not make a summer". Much more extensive numerical study and comparison with other methods using, for example, the test problems in [11] is needed. Alternatively, one could apply a QC-type technique to a practical problem to determine its effectiveness, and indeed one of the objectives of this article is to seek collaboration on such an application.

We should also mention here the cautionary experience reported in Carter [2] where the number of iterations grows exponentially with a measure of inaccuracy in gradient evaluation, but note also that in [2] function values are inexact, and directional derivative information is *not* exact (upto truncation error in a finite difference estimate) in *any* direction.

4. DISCUSSION OF QC METHOD

4.1 Refinements: Strategies involving updating a quasi-gradient over large steps using a non-symmetric Broyden-type update formula is interesting theoretically but did not prove to be effective (the gradient is not constant even on a quadratic, in contrast to the Hessian); neither did updating the metric using quasi-gradients vectors (they are themselves estimates). After winnowing out these options (an exploration facilitated by Matlab), the simpler and more elegant algorithm of Section 3 emerged as more promising, and it can be refined in a veriety of ways, some of which are listed below:

1. If very little progress is made in the line search from $\underset{\sim}{x}$ along a direction, say d_-, then the set of vectors in the previous block of vectors used at x_- and their associated directional derivatives could be included in the set $e_1, \ldots e_k$ used at x. Clearly this will eventually yield a good approximation to the gradient if performed repeatedly.

2. Additionally or alternatively, the quasi-gradient $\bar{g}$ can be chosen to minimize the euclidean norm of $\bar{g} - \bar{g}_-$, where $\bar{g}_-$ is the quasi-gradient estimate at x_-. It is easy to construct an example where this could result in a non-descent direction at x, bringing the lemma of section 2.1 directly into play.

3. A basis other than the coordinate vectors can be employed. In addition, one or more randomly chosen directions, with directional derivatives estimated by difference along them, could be added to the set used to estimate the quasi-gradient at each iterate. Introducing some randomization into the algorithm could be very useful.

4. A line search along the direction defined by joining iterates at the beginning and end of a full cycle of K blocks, as in some pattern search methods, could be a useful acceleration step.

5. If the set of vectors used to estimate a quasi-gradient at an iterate does not yield a direction of descent and must be augmented then well-known computational linear algebra techniques can be employed to obtain the new quasi-gradient efficiently.

4.2 The QC Relation and Diagonal Updating: Let us define the *quasi-Cauchy (QC) Relation* to be the weak secant relation $sM_+s = b$ of [6] with the added restriction that M_+ must be *diagonal*. Henceforth let us write this quasi-Cauchy relation as

$$s^T D_+ s = b \equiv (g_+^T s - g^T s). \qquad (3)$$

We have noted that the matrix D_+ of section 2.3 does not satisfy this QC relation. However, a whole range of variational problems can be formulated where (3) is a constraint that must be satisfied, and they lead to different updates of D to D_+, some of which preserve positive definiteness. *There are some interesting results concerning the updating of diagonal matrices that can be obtained,* and they may be of practical use in other settings, for example, preconditioning CG algorithms as well as QCG extensions. For those diagonal updates that do not preserve positive definiteness, one might also consider a model-based QC approach (see [12]).

A useful background reference for diagonal updating is Gilbert and Lemarechal [9].

4.3 Unifying Framework: The QC approach broadens, in an attractive way, the Newton/Cauchy framework discussed in [12]. *Thus even if the QC method does not prove to be competitive, at the very least it will provide a stepping stone and a natural transition into the subject of derivative-free methods (pattern search, simulated annealing, genetic algorithms, etc.) that deserve a larger role in the optimization curriculum.*

The topics discussed in Sections 3 and 4.1-4.3, convergence analysis and connections/contrasts (Gauss-Seidel when the function is quadratic, block cyclic coordinate descent) are currently under investigation and will be described in a technical report at a later date.

References

[1] Barzilai, J. and Borwein, J.M. (1988), "Two-point step size gradient methods", *IMA. J. Numerical Analysis*, 8, 141-148).

[2] Carter, R.G. (1993), "Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information.", *SIAM J. Scientific Computing*, 14, 368-388.

[3] Cauchy. A. (1829), "Sur la determination approximative des racines d'une equation algebrique ou transcendante", *Oeuvres Complete (II)*, 4, 573-607. Gauthier-Villars, Paris, 1899.

[4] Davidon, W.C. (1959), "Variable metric method for minimization", Argonne National Laboratory, Report ANL-5990 (Rev.), Argonne, Illinois (reprinted with a new preface in *SIAM J. Optimization*, 1, 1991).

[5] Dennis, J.E. and Torcson, V. (1991), "Direct search methods on parallel machines", *SIAM J. Optimization*, 1, 448-474.

[6] Dennis, J.E. and Wolkowicz, H. (1990), "Sizing and least change secant updates", CORR Report 90-02, Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada.

[7] Ermoliev, Y. and Gaivoronski, A.A. (1994), "Stochastic quasigradient methods", *SIAG/OPT V&N*, 4, 7-10.

[8] Fletcher, R. and Powell, M.J.D. (1963), "A rapidly convergent descent method for minimization", *Computer Journal*, 6, 163-168.

[9] Gilbert, J.C. and Lemarechal, C. (1989), "Some numerical experiments with variable storage quasi-Newton algorithms", *Mathematical Programming*, 45, 407-436.

[10] Griewank, A. (1989), "On automatic differentiation", in *Mathematical Programming: Recent Developments and Applications*, M. Iri and K. Tanabe (Eds.), Kluwer Academic Publishers, Dordrecht, 83-108.

[11] Moré, J.J., Garbow, B.S. and Hillstrom, K.E. (1981), "Testing unconstrained optimization software", *ACM Trans. on Mathematical Software*, 7, 17-41.

[12] Nazareth, J.L. (1994), *The Newton-Cauchy Framework: A Unified Approach to Unconstrained Nonlinear Minimization*, LNCS 769, Springer-Verlag, Berlin.

[13] Powell, M.J.D. (1991), "A view of nonlinear optimization", in J.K. Lenstra, A.H.G. Rinnooy Kan and A. Schrijver (Eds.), *History of Mathematical Programming*, North-Holland.

[14] Shor, N.Z. (1985), *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, Berlin.

[15] Tseng, P. (1995), "Fortified-descent simplicial search method: a general approach", preprint.

****

BULLETIN BOARD

CONFERENCE ON NETWORK OPTIMIZATION PROBLEMS

A conference on network optimization problems will be held at the Center for Applied Optimization, University of Florida from February 12-14, 1995. The conference will bring together researchers working on many different aspects of network optimization: algorithms, applications, and software. The conference topics include diverse applications in fields such as engineering, computer science, operations research, transportation, telecommunications, manufacturing, and airline scheduling. Since researchers in network optimization come from many different areas, the conference will provide a unique opportunity for cross-disciplinary exchange of recent research advances as well as a foundation for joint research cooperation and a stimulation for future research.

Advances in data structures, computer technology, and development of new algorithms have made it possible to solve classes of network optimization problems that were recently intractable. For example, recent advances have been made in techniques for solving problems related to airline scheduling, satellite communication and transportation, and VLSI chip design. Computational algorithms for the solution of network optimization problems are of great practical significance.

The conference will be held at the Center of Applied Optimization, University of Florida, Gainesville, Florida. All presentations are invited. A collection of refereed papers will be published in book form by Kluwer Academic Publishers. Details will be available at a later time.

For further details, please contact one of the conference organizers: Bill Hager (hager@math.ufl.edu), Don Hearn (hearn@ise.ufl.edu) and Panos Pardalos (pardalos@ufl.edu).

CONFERENCE ANNOUNCEMENT: ICCP-95

An International Conference on Complementarity Problems: Engineering & Economic Applications, and Computational Methods will be held from November 1-4, 1995 at the Homewood Campus of The Johns Hopkins University, Baltimore, Maryland, U.S.A.

The conference is organized by: Michael C. Ferris, University of Wisconsin at Madison (ferris@cs.wisc.edu) and Jong-Shi Pang, The Johns Hopkins University (jsp@vicp.mts.jhu.edu)

The conference will bring together, for the first time, engineers, economists, industrialists, and academicians from the U.S. and abroad who are involved in pure, applied, and/or computational research of complementarity problems, to present and discuss the latest results in this subject, and to offer suggestions for collaborative research and further development of the field.

The conference will last for 4 days, consisting almost entirely of invited presentations. There will be a small number of selective contributed talks and the conference is limited to 100 participants (including the speakers). A refereed volume of proceedings of the conference will be published.

There are three major themes of the conference: engineering applications, economic equilibria, and computational methods. Each theme will be represented by experts in the area.

Contact one of the organizers for further details if you are interested in participating at the conference or in contributing a paper for possible presentation.

CONFERENCE ANNOUNCEMENT: HIGH PERFORMANCE SOFTWARE

A short conference on "High Performance Software for Nonlinear Optimization: Status and Perspectives". will beheld from June 22-23, 1995 (following the Workshop in Erice on Nonlinear Optimization), and hosted by Centro di Ricerche per il Calcolo Parallelo e i Supercalcolatorim (CPS), a joint research center of the CNR (Consiglio Nazionale delle Ricerche) and the University of Naples "Federico II".

The conference, which will be held in Capri, Italy, will focus on the current state of optimization software, mainly referring to the high performance computing aspects. The presentations will provide an authoritative overview of the field, including its algorithmic developments, current software and applications, and future perspectives. There will be a series of lectures given by invited speakers including: P.Matstoms, J. J. Moré, W. Murray, P.M. Pardalos, M. Resende, P.L. Toint, and S. Zenios. There will also be a selection of contributed papers.

For more information or for registration, send an e-mail message to hpsno@matna2.dma.unina.it or a letter to Prof. Almerico Murli (Head of the CPS): Universita' di Napoli "Federico II" Dipartimento di Matematica e Applicazioni, Complesso Monte S.Angelo, ed. T, Via Cintia 80126, Napoli. Italy. E-mail: murli@matna2.dma.unina.it.

SUBSCRIBER HELP INFORMATION:

- **opt-net** (Optimization):
  opt-net-request@zib-berlin.de with Subject "help" and message body "EOI".

- **na-net** (Numerical Analysis):
  na.help@na-net.ornl.gov

- **pol-list** (OR/MS Practice):
  pol-request@silmaril.smeal.psu.edu with message body "help".

SELECTED UPCOMING ARTICLES FOR SIAM J. OPTIMIZATION

Why Broyden's Nonsymmetric Method Terminates on Linear Equations *Dianne P. O'Leary*

A New Infinity-Norm Path Following Algorithm for Linear Programming *Kurt M. Anstreicher and Robert A. Bosch*

A Potential Reduction Algorithm with User-Specified Phase I–Phase II Balance for Solving a Linear Program from an Infeasible Warm Start *Robert M. Freund*

An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima *P. Gilmore and C. T. Kelley*

Indefinite Trust Region Subproblems and Nonsymmetric Eigenvalue Perturbations *Ronald J. Stern and Henry Wolkowicz*

A Reduced Hessian Method for Large-Scale Constrained Optimization *Lorenz Biegler, Jorge Nocedal, and Claudia Schmid*

A Robust Trust-Region Algorithm with a Nonmonotonic Penalty Parameter Scheme for Constrained Optimization *Mahmoud El-Alem*

A Class of Trust Region Methods for Nonlinear Optimization Problems *A. Sartenaer*

Ladders for Traveling Salesmen *Sylvia C. Boyd, William H. Cunningham, Maurice Queyranne, and Yaoguang Wang*

On the Convergence of Fenchel Cutting Planes in Mixed-Integer Programming *E. Andrew Boyd*

Proximal Decomposition on the Graph of a Maximal Monotone Operator *Philippe Mahey, Said Oualibouch, and Pham Dinh Tao*

On the Simulation and Control of Some Friction Constrained Motions *Roland Glowinski and Anthony J. Kearsley*

Global Convergence of a Long-Step Affine Scaling Algorithm for Degenerate Linear Programming Problems *Takashi Tsuchiya and Masakazu Muramatsu*

On Eigenvalue Optimization *Alexander Shapiro and Michael K. H. Fan*

Incorporating Condition Measures into the Complexity Theory of Linear Programming *James Renegar*

The Linear Nonconvex Generalized Gradient and Lagrange Multipliers *Jay S. Treiman*

Taylor's Formula for $C_{k,1}$ Functions *Dinh The Luc*

Nonpolyhedral Relaxations of Graph Bisection Problems *Svatopluk Poljak and Franz Rendl*

A Sequential Quadratic Programming Algorithm Using an Incomplete Solution of the Subproblem *Walter Murray and Francisco J. Prieto*

Data Parallel Quadratic Programming on Box-Constrained Problems *Mike P. McKenna, Jill P. Mesirov, and Stavros A. Zenios*

Local Convergence of SQP Methods in Semi-infinite Programming *G. Gramlich, R. Hettick, and E. W. Sachs*

Faster Simulated Annealing *Bennett L. Fox*

CONTRIBUTIONS TO THE V&N

The next issue (Fall, 1995) will include essays by Dimitri Bertsekas (MIT; preview of new book and a discussion of optimization trends) and Paul Frank (Boeing; on expensive function optimization using interpolation models).

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. *A contribution on methods discussed in the feature article (page 1) for the next issue would be especially appreciated.*

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 deadline for the next issue is October 1, 1995.

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