Efficient Optimization through Response Surface Modeling: A GROPE Algorithm

A Dissertation

Presented to

the Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Systems Engineering

Submitted by

John F. Elder IV

May 1993

Abstract

For many real-world optimization applications, such as finding control parameters for a physical or simulated system, responses to test probes are expensive to obtain and gradient information is unavailable. An algorithm to address this class of GROPE problems (Global ^{d}Optimization when Probes are Expensive) is developed and implemented, which is very efficient in the number of function evaluations required. The multi-dimensional search algorithm generalizes a 1-dimensional procedure in which a "random walk" or fractal form is assumed for the response surface (Kushner 1964). A piecewise model of the emerging surface is maintained, with each region (Delaunay simplex) having linear expectation and quadratic variance. The model is queried for promising new search locations; i.e., the design points (given all known results) most likely to exceed the current performance goal. A "parameter-free" implementation of the algorithm demonstrates results superior to those of other techniques in the literature on a suite of standard two-dimensional test functions.

APPROVAL SHEET

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Systems Engineering.

_____

John F. Elder IV

This dissertation has been read and approved by the examining committee:

_____

D. E. Brown, Advisor

_____

W. T. Scherer, Committee Chair

_____

J. Aitchison

_____

J. S. Salowe

_____

N. Nandhakumar

Accepted for the School of Engineering and Applied Science:

_____

Dean, School of Engineering and
Applied Science

May, 1993

Acknowledgements

This work is dedicated to my dear wife, Elizabeth.

My years of (often intense) study and research here at the University of Virginia have been remarkably pleasant and rewarding. The chance to thoroughly pursue challenging research under the tutelage of great teachers, and in the stimulating company of able fellow-students, has been nothing short of a blessing from God. I am deeply grateful also to my committee, whose comments and discussions have improved my research and its presentation, and whose willingness to accommodate my recently accelerated schedule has made graduation possible! In particular, I commend to those following, the fine courses offered by Drs. Brown, Aitchison, Salowe, Scherer, and Nandhakumar.

From my youth, I have yearned to be an inventor and (though not matching the exploits of a Tom Swift) I've found something of a calling in devising algorithms for addressing practical engineering and scientific problems. That I could have the chance to strengthen my understanding of the essential underlying issues, while also feeding my (growing) family, is largely the work of two men: Don Brown, and Mark Finn. Don has arguably been the perfect advisor: steady with support, reasonable in demands, and an exemplar of high scholastic standards. Mark too, could scarcely be improved upon if I were designing a boss: this work would not have been possible without his belief in me and his regular support of my research over the years.

Lastly, I am deeply indebted to my wife, Elizabeth, whose love is a constant encouragement. Her dedication to raising the delightful little people whom God has entrusted to us -- Catherine, Jonathan, and Anna -- has made these times happy indeed.

Table of Contents

List of Figures

List of Tables

List of Symbols

d.....dimension

.....real space

A.....search space $\subset$ ^{d}

$\underline{x}$.....location *d*-vector

y.....response value

F().....response function: $\underline{x} \rightarrow y$

Y_g.....goal for *y*

N.....number of probes

L .....Lipschitzian (slope) constant

ε.....location error tolerance

$\mu()$ .....content-measuring function

$\hat{\mu}()$.....content ratio function

$\bar{y}()$.....mean function

$\sigma^2()$.....variance function

p.....proportion (or probability; note context)

c.....mean square variation in *y* w.r.t. $\underline{x}$

$D^2()$ .....squared standardized distance function

Y_{min}, Y_{max}, $Y_{max}(k)$.....minimum, maxiumum, k^{th} maximum sampled *y*

P_{max}.....maximum allowed number of probes

γ .....Y_g attentuation

b_{0j}, b_{1j} .....lower, upper bound on x_j

v_{ij}.....x_j for i^{th} vertex

X, **Z**.....original space, log-ratio transformed space

1. Introduction

A general-purpose optimization procedure can act as a "safety net" in engineering and scientific research, catching applications that stray off the narrow path of assumptions required to employ better-understood techniques, such as *regression* (linear least squares estimation) and *linear programming* (linear least absolute values estimation). By sacrificing the efficiency possible under rigid assumptions, one can address very general problems of parameter optimization, where a response function, $F(\underline{x})$ is to be minimized

$$\underline{x}^* = \text{argmin } [F(\underline{x})] \tag{1}$$

subject to some bounds on the control vectors, $\underline{x}_i$

$$\underline{x}_i \in \mathbf{A} \tag{2}$$

through "intelligent trial and error"; i.e., through guided *probing* of the system.

In this research, we develop and implement a new algorithm designed to guide such probing efficiently; that is, to obtain "good answers", with high confidence, in as few probes as possible. To obtain this "running" efficiency however, the algorithm spends considerable time "thinking". In fact, in its present form, the required overhead is likely to be more considerable than under any other technique. Hence, the appropriate applications are relatively low-dimensional ($d \leq 6$ or so), and have response functions which are *expensive* to evaluate, thereby justifying probe frugality.

The latter property at least, is common in practice. Though the usual academic test functions are simple equations requiring only a few lines of code to implement, industrial optimization applications often involve performing a physical experiment, or running a simulation consisting of thousands of lines of code. In our experience, such problems include identifying robust and accurate nonlinear parameters for automatic fighter aircraft

control reconfiguration (e.g., Elder and Barron, 1988), stock portfolio selection (e.g., Elder and Finn, 1991), and missile guidance (e.g., Barron et al., 1990).

As such functions, $F(\underline{x})$, are not analytic, no gradient information, $F'(\underline{x})$, is available. Also, there may exist many local minima, several of which could be close to the overall minimum. Restricting attention to *real* control parameters, the applications to be addressed are in what we call the *GROPE* class: Global R^d Optimization when Probes are Expensive.

For example, in the guidance application mentioned, the response function to be optimized was a complex (and user-defined) function of impact conditions: location, angles of incidence, velocity, and angles of attack. The control parameters were a relatively abstract vector of initial Lagrange multipliers, $\underline{\lambda}_0$, (typically of size 6) controlling, in subtle ways, the guidance tradeoffs made between each of the rotational and translational degrees of freedom available. Each probe of the system, to associate a response with a trial vector, $\underline{\lambda}_{0_i}$, required integration of the equations of motion of a simulation from release until impact -- a process which took 7 to 8 minutes on a (rented) VAX computer in the project's early days. (In retrospect, the need for an efficient algorithm capable of exploring complex regions with human-like skill in dimensions beyond those one could visualize, was indelibly impressed upon the author by years of late nights on this type of project!)

A practical optimizing algorithm should therefore be *global* (not easily trapped in local minima), *direct* (employ only probe responses), and efficient (as probes involve considerable calculation). It should also provide some measure of confidence in the result(s).[1] Due to the expense of probing in GROPE problems, we further assume that even pseudo-gradient information, such as is possible through finite difference techniques, is not practical to obtain.

---

[1]The latter two properties, not strictly required, are nevertheless important in practice, as the defining parameters for a search problem are often adjusted as long as time permits to explore alternative scenarios.

The types of optimization algorithms appropriate for this important class of problems are enumerated and critiqued in Chapter 2. There, the class of algorithms which model the response surface to improve estimation are highlighted as demonstrating particular promise by the metric of minimizing probes. Chapter 3 describes the elegant 1-dimensional algorithm of Kushner (1964), wherein probes are chosen to maximize the probability of exceeding a goal response value under a random-walk model of the response surface. Chapter 3 extends Kushner's results slightly (with details in the Appendix), and then generalizes the algorithm to multiple dimensions, using the Delaunay triangulation to divide the search space into independent regions. The algorithm is designed to be as user-friendly as is practical, requiring of the user only a "black box" routine implementing the function, a list of univariate bounds, and a probe limit. A proof of convergence concludes Chapter 3.

Chapter 4 describes the important generalization and implementation issues, and Chapter 5 presents the results of exercising the algorithm on a suite of standard two-dimensional test functions from the literature. The new GROPE algorithm, with its minimal input requirements, is shown to be slightly better than its leading competitor -- a similar model-based algorithm by Cox and John (1992) -- and, with it, to dominate the alternatives.[2] The program is shown to do best on functions where local, rather than global, structure dominates in the determination of important minima. Chapter 5 also briefly reveals the effect of dimension and the number of probes on overhead time. Chapter 6 concludes with a summary of the contributions of the research and an overview of important future research topics.

---

[2]Better performance is also revealed in cases where some information about the function is known *a priori*.

2. The GROPE Problem

2.1 Solvability

Strict global optimization is, in general, an unsolvable problem for spaces that cannot be exhaustively evaluated (Torn and Zilinskas, 1989). It is not usually possible to find singular global minima occurring at isolated points, but even ruling out such "poles in the desert" is not sufficient for obtaining solvable problems (op cit.); in fact, finding the optimal (minimal) value F^* or its location $\underline{x}^*$, is unsolvable even for functions, F, with continuous derivatives of any degree in the search region. That is, there exist functions for which the global minimum may differ arbitrarily much from the minimum value found after a finite number of steps.

For practical applications therefore, the search goal must be to: Find as good an answer (or set of answers) as possible given the resources (e.g., compute time) available.[3] Since one is always "stopping early", the likelihood of a better answer being "out there somewhere" is important information for the analyst. For the relaxed problem of identifying a neighborhood, or level set, of the global minima (of arbitrarily small but positive content), Solis and Wets (1981) have shown that a wide range of procedures, given infinite time, can guarantee a solution. Of practical interest are those which are generally non-dominated in the performance space of time and quality; that is, the methods likely to provide the best F^*-to-time tradeoffs for real applications. Dominance in expected-case or worst-case scenarios will not be formally addressed here, but results will be compared with those of competing procedures on representative functions to begin to reveal the relative properties of the new algorithm.

---

[3]Henceforth, in the context of GROPE problems, "optimization" will have this default meaning.

2.2 Types of Global Algorithms

Many classical search techniques are local, such as the Newtonian methods, and thus appropriate only for unimodal response surfaces. Many also require gradient information. Still, there remain several direct algorithms potentially appropriate for exploring multi-dimensional, multi-modal real surfaces. Torn and Zilinskas (1989) discuss the classifications many authors have made of global optimization algorithms, and argue for six basic types:

1. Covering methods
2. Random search
3. Clustering methods
4. Generalized descent
5. Methods approximating the level sets
6. Methods approximating the objective function

They also further define class 1 as a "method with guaranteed accuracy", class 2-4 as "direct methods" (here, meaning they employ only local information; i.e. probe results), and classes 5-6 as "indirect" (they further construct a global model with the responses for use by the search). However, covering methods are *guaranteed* only under assumptions (e.g., slope limits) more restrictive than present with GROPE problems. Also, the gathering and classifying of probe results performed by clustering methods seems more reminiscent of the modeling methods (5 and 6) than the random and descent methods (2 and 4). We suggest instead two groupings, based on whether the method attempts to get a "big picture" of the pattern of emerging probe responses, or is more local in its view.

Big Picture: covering, clustering, and modeling approaches (1, 3, 5, and 6);

Blinders On: random and descent methods (2 and 4).

Techniques falling into each of these categories, to varying degree, can be appropriate for GROPE problems. The algorithm introduced here is in class 6, but has some similarity to those in class 1. Before describing its development, we now briefly survey the existing techniques, framed by the above classification.

2.3 Covering Methods

Covering methods operate by successively eliminating subregions of the search space which cannot contain the optimum point. In ^{d} this is made possible by employing bounds on the gradient of F; e.g., a Lipschitz constant L, s.t.

$$| F(x_1) - F(x_2) | \leq L \, \| x_1 - x_2 \| \qquad x1, x2 \in \; ^{d} \tag{3}$$

That is, the magnitude of response difference for two points is bounded by L times their separation, as measured by some norm, ||*||. (Note that, in multiple dimensions, the variables must be properly scaled; see, e.g., (Kruskal and Wish, 1978).) Estimating L can be especially difficult for simulations; yet, if the estimate, $\hat{L} >> L$, the search will take much longer than necessary, and if $\hat{L} < L$, the minimum can be missed.

The simplest algorithms are *passive*, having a fixed sample pattern. A straightforward cubic grid requires

$$N \geq \left(\frac{\sqrt{d}}{2\delta} \right)^{d} \tag{4}$$

samples (Torn and Zilinskas, 1989), where δ is the desired accuracy in each dimension (which can be related to L). Sobol (1979) introduced grid patterns with no "shadow effect"; i.e. no coincidence of points when projected onto an axis. In fact, arbitrary

projections of such grids onto any subspace are asymptotically uniform -- even more so than pseudo random grids, though this distributional advantage decreases with *d*.[4]

As the samples required for these passive grids grows severely with dimension, there is interest in *active* (or "sequential") methods, which can reduce sampling requirements by adapting the pattern according to probe results. A multi-stage grid search (e.g. Niederreiter and Peart, 1986) concentrates ever finer grids around the best probe from the previous stage. This process can converge relatively quickly but risks losing the global minimum.

A more reliable technique employs *branch and bound* (e.g., Horst and Tuy, 1987); yet, use of any such method introduces the algorithmic requirement of employing recursive partitions. That is, to make the algorithm recursively feasible, the space remaining after exclusion of a region must have the same structure as before. Therefore, spherical regions (natural to a Euclidean norm) are problematic, and instead cubes, rectangles (e.g., Jones et al., 1992), or simplices (e.g., Wood, 1985; Perttunen, 1991; Elder, 1992) are employed to tessellate the search space. Still, such methods tend to accumulate long lists of sub-regions to explore -- a "bookkeeping" task that worsens rapidly with *d*. (The use of such simplices by the algorithm introduced here is the chief reason for its practical limitation to a moderate dimension, *d*, of 10 or so.)

Covering methods using *interval arithmetic* (e.g., Ratschek, 1985) usually call for derivatives and are hard to generalize to multiple dimensions. *Integral* methods (e.g., Pinkus, 1968) offer closed-form solutions for a narrow class of analytic functions. The Lipschitzian techniques are the only practical GROPE method, but results depend strongly on the tightness of the bound $\hat{\Delta}$. For example, Mladineo (1986) reports on minimizing

$$F(x,y) = 4xy \sin(4\pi y) \tag{5}$$

---

[4]Torn and Zilinskas (1989) refer to a 1981 Russian book or report, *Choice of optimal parameters in problems with many criteria*, by Sobol and Statnikov for these grid results.

in the unit square. Respective $\hat{L}$ estimates of {39, 28, 25} required probe sequences of length {833, 119, 44} for a given accuracy.

Other than the difficulty of estimating L (especially with response functions defined by computer simulations rather than equations), bounds on the first and second derivatives may not be the best way to classify functions. In practice, the maximum slope is often far from the minimum response value, making its determination moot in the areas of greatest interest (unless the estimate is adapted to each region -- an added complexity). Further, the natural Lipschitzian model (lower bound) for the response surface is a collection of cones (centered at x_i, with heights $F(x_i)$, for $i = 1, N$) sloping downward with rate $\hat{L}$, until intersecting with a neighboring cone. This complex surface is multimodal, and no efficient algorithms for its exploration are known for $d \geq 4$ (Torn and Zilinskas, 1989), thus requiring further approximation.

2.4 Random Methods

Techniques involving random search are widespread due to the ease of programming and adapting them to heuristic ideas of the users. The consensus of researchers appears to be that the methods are robust (not sensitive to unusual features of the problem) but can require huge numbers of probes, making them unsuitable for many GROPE problems. Still, they provide a useful baseline for comparing techniques.

Probe requirements worsen dramatically with dimension. For example, with N uniformly distributed probes, the probability, p, that one of the probes is in the local neighborhood of the global minimum, $\underline{x}^*$ (a fraction f of the total space) is

$$p = 1 - (1 - f)^N \tag{6}$$

Thus, to be 95% sure of hitting the right 1% of the space (e.g., the correct decile in each axis of a two-dimensional problem) takes about 300 probes. If the goal is to minimize the

expected Euclidean distance (L_2 norm) from a grid point to a random location (x^*), uniform random samples are better than a cubic grid when $d \geq 7$ (Andersson and Bloomfield, 1975). However, if the maximum separation of coordinate values is to be minimized (L norm), a pseudo-cubic grid is always preferable (Aird and Rice, 1977).[5]

Adaptive random search methods combine random sampling with local optimization. The best known is the *multistart* algorithm which simply begins local searches from several random points. An alternative is to perform searches along randomly chosen lines. Adaptation can be structural or parametric. *Structural* adaptation occurs when an algorithm switches search strategies to exploit perceived properties of the response surface. This occurs trivially in many methods which transition to a local search strategy when sample returns suggest they are in the vicinity of a minimum, but it is a more thoroughgoing principle in a few algorithms -- e.g., *Grope* (Flood and Leon, 1965)[6] and *OmniSearch* (Hess and Abbott, 1988), which can alternate between several specialized strategies.

Parametric adaptation can be as simple as reducing the sampling range with time. For example, the algorithm of Barron (1968) takes random draws from a Gaussian distribution, with shrinking □, centered at the current best point. When a better location is identified, it then performs a simple search along the line defined by the two leading points (with step size adapting to the results).

Simulated annealing algorithms (e.g., Metropolis et al., 1953; Aarts and Korst, 1989) also sample in the vicinity of an "accepted" point but, with decreasing probability p, that point is not the current best. This (gradually less likely) acceptance of a worse point is important

---

[5]Apparently however, it is still an open question as to how *optimal* representatives of random and deterministic sampling compare (Torn and Zilinskas, 1989).

[6]GROPE is a popular search name. We have used it (as an acronym) in predecessor model-based algorithms twice: Guided Random Optimizer of Performance Error (employed in Elder and Barron, 1988), and Global Regression of Probe Evaluations (Elder, 1991).

for escaping the *basins* (regions of attraction of) local minima. Still, though simulated annealing asymptotically converges to the true minimum with probability one (Faigle and Schrader, 1988), a huge number of probes are generally required for a good solution (e.g., Aluffi-Pentini et al., 1985).

A *non-parametric* type of adaptation can be envisioned in which the sampling density is made roughly inverse to the emerging response surface; e.g., high where the surface is low. A baseline (or prior) uniform sampling distribution could be modified by probe results to attract samples to areas which appear promising and repel them from others. That is, positive *kernels* of probability mass could be centered at good locations, negative at bad, with the global "sea level" adjusted to maintain a constant total mass. The mass and/or widths of the kernels could further depend on the local response values. A method where the global sample distribution is adapted to the probe responses in a somewhat similar manner is by McMurty and Fu (1966). There, after an initial uniform random sampling of the space, subsets are selected for further investigation according to the aggregate performance of the probes within them. In particular, the probability of sampling within a subset A_j is proportional to the average value of

$$F(x_i)^{-\alpha}, \; x_i \in A_j \tag{7}$$

for positive $F(x)$, and positive constant, α.

Other search mechanisms in which randomness plays an important role include those based on evolutionary theory; most notably, *Genetic Algorithms* (e.g., Holland, 1975; Goldberg, 1989).

2.5 Clustering Methods

Cluster methods attempt to avoid a pitfall of multistart -- repeated discovery of known minima -- by defining regions of attraction for each through clustering. By covering the

search space with clusters with (presumed) known end-results, probes may be conserved. Torn and Zilinskas (1989) describe several such methods. The key steps are

1) Sample points in a region of interest;
2) *Concentrate* the sample to obtain groups around the local minima; and
3) Recognize these groups with clustering.

The distinctive second step is performed either by identifying points with low function values (e.g., Becker and Lago, 1970; see Price, 1978), or by "pushing" each chosen point towards its local minimum with a few steps of a local algorithm (e.g., Torn, 1977).

Clustering (like optimization, a "black hole" for researchers!) abounds in techniques (hierarchical or non-, agglomerative or divisive, etc.). In this search application however, there are two freedoms not found in the basic *d*-dimensional clustering problem:

1) The function value, *F(x)*, is informative; for example, low values may be good "seed points" for iterative clustering techniques; and
2) New points may be introduced (new locations probed) during the analysis.

Clustering algorithms can be efficient; for example, a Bayesian version by Betro and Rotondi (1984) called for only a few duplicate local searches on a set of standard test problems. The advantage stems from sharing information between probes; that is, in forming something of a picture of the emerging pattern of probe results from the individual returns. (This method for improving efficiency is even more fully exploited by the modeling techniques, discussed in Sections 2.7 and 2.8.)

Local grouping of probe information can be done in other ways as well. For example, Timmer (1984) describes a method based on *nearest neighbors* (e.g., Cover and Hart, 1967). The technique starts only a finite number of local searches in infinite time, and provides an optimal Bayesian estimate of the number of minima remaining to be found.

2.6 Methods of Generalized Descent

There are four types of descent algorithms for global (or perhaps, *multi-local*) optimization:

1) Multistart of local techniques (discussed above);

2) *Trajectory* methods, which modify the equation of local descent;

3) *Penalty* methods, which modify *F(x)* to avoid known (found) local minima; and

4) *Simplex* methods (e.g., Walters et al., 1991), which manipulate a collection of *d+1* points, and tend to "descend by not ascending".

Trajectory methods typically employ a mechanical analogy of a ball rolling on the response surface (e.g., Incerti et al., 1979). Set down at an arbitrary location, the ball will trace out a path to a local minimum, but because of its inertia, some shallow minima will be passed over. However, this method, as well as non-inertial trajectory techniques (e.g., Branin, 1972), relies critically on first and second-order derivative information, which is usually either too expensive or too noisy when estimated from probes of GROPE problems. Gomulka (1978) reports that it offers no advantage over the simpler multistart approach.

Penalty methods avoid the "multi-visit" pitfall of multistart by artificially increasing the response function values in their vicinity. When F is known analytically, it may be possible to "cancel its zeros with poles", but penalty methods can work for empirical functions as well. For example, the "filled function" method of Ge and Qin (1987) adds an exponential symmetric kernel to F, centered at each local minima, x_i, "filling" that depression locally. A similar idea, but different penalty form, is employed in the "tunneling" methods (e.g., Levy and Montalvo, 1985). Still, though old minima are easy to exclude, one must take care not to introduce false new ones. Further, the penalized functions can easily become rather featureless and "flat", making them difficult to work with after the initial batch of minima are found (Torn and Zilinskas, 1989).

Price (1978) describes a simplex method which tends to cluster the $d+1$ points around the best minima. But the method due to Nelder and Mead (1965) (based on (Spendley, et al., 1962)) is one of the most widely used direct search methods. (For instance, Torczon (1989) reports that over half the calls received by the support group for the NAG software library concern their version of that program. Also, it comes highly recommended in the popular book and software package *Numerical Recipes* (Press et al., 1988).) Their *downhill simplex* algorithm consists of a sequence of heuristic steps, most of which exchange the worst point of the simplex for a new probe; in particular, they reflect the simplex through the *face* formed by the other points. Typical reflection magnitudes, depending on the latest results, are -0.5 (contraction), 1, and 2 (reflection and expansion). Occasionally, the simplex contracts in size toward the best point (requiring d new probes).

Some attempt to conserve the volume of the simplex is made to avoid degeneracy, but problems have long been known to exist with its use, especially in high dimensions. Torczon (1989) (who introduces a simpler and more robust version of the algorithm for parallel machines) references several fruitless attempts to prove that the Nelder-Mead algorithm converges in general, and reports experimental results where it converged to *local* minima -- even for simple functions -- as early as $d = 8$.

Still, despite these weaknesses, the Nelder-Mead algorithm remains very popular with users,[7] as it acknowledges the GROPE problem characteristics of having noise and not derivatives (rather than the other way around!). Also, it seems to be easy to understand and program. Clearly, another derivative-free algorithm with superior robustness and speed will attract user interest, especially if it also "makes sense" heuristically.

---

[7]For instance, a review of Nelder-Mead citations in the *Science Citation Index* by Dennis and Torczon (1991) revealed that both the number and range of journals involved has grown every year since the algorithm's appearance in 1965. The 215 citations in their latest year (1989) were about twice that of the leading quasi-Newton method (Fletcher and Powell, 1963).

2.7 Methods Approximating the Level Sets

Techniques for modeling the level sets, $L(t)$, for search region A,

$$L(t) = \{x \in A \mid F(x) \leq t\} \tag{8}$$

are not very common, and seem to all stem from work by Chichinadze (e.g., 1969, 1991). Chichinadze proposed estimating $\mu(L(t))$ from finite samples, where μ is a measure of volume or *content* (e.g., Lebesque), then modeling the proportions

$$\phi(t) = \frac{\mu(L(t))}{\mu(A)} \tag{9}$$

with orthogonal polynomials. Lastly, x^* is estimated by determining the root, $\phi(t) = 0$.

The technique involves several stages of approximation; also, extrapolation to estimate the tails of a distribution is typically not accurate (e.g., Dixon and Szego, 1978). However, some similar ideas have seen use for establishing stopping conditions -- a less critical problem than locating x^*. They involve estimating the distribution parameters of either the

1) responses, to infer its peak value (e.g., Hartley and Pfaffenberger, 1969), or

2) extreme values. (For example, Derigs (1985) employs a Weibull distribution to get a bound for *probabilistic branch and bound*. In a different context, Pannullo (1992) discusses estimation of parameters for *Gumbel types*.)

Note however, that for the distributions to be random (in line with the theories) the probes should be randomly located. For all but unguided techniques, this assumption is decreasingly valid with time.

2.8 Methods Approximating the Objective Function

It is helpful, at this stage, to liken global optimization to depth-sounding (Figure 1). One probes for the depth (evaluates the response function) at a location (set of parameter values) in pursuit of the deepest spot (global minimum). *Model-based* search methods build up a picture of the ocean floor from probe results, to home in on the goal as quickly as possible, and to have some confidence that a (reasonably) deeper point is not "out there somewhere".

The model of the response surface is interrogated for a location to probe. Should analytic solutions prove too difficult, *internal model search* can be employed; that is, the model (as a rapid surrogate for the true score function) can be sampled at candidate probe locations to determine the most promising one. New results update the model, and the cycle continues.

Though the overhead computations are extensive, a good model can drastically reduce function evaluations. If probes are moderately expensive, the time spent “thinking” will be more than compensated for by the time saved “running”. Further, the essential estimation stage shifts the algorithmic emphasis from the *minimax* optimality sought by many techniques, to the more rational *average* optimality metric (Torn and Zilinskas, 1989).

Some early model-based approaches are described, and axiomatically defended, in (Zilinskas, 1982). Crude linear or quadratic regression models were employed in (Elder and Barron, 1988), though with only a slight effect on the underlying guided random search: the simple global models were not responsive enough to variations in local topology, yet too responsive to outliers on the fringe of the search space, due to use of the squared error criterion in modeling. Significant refinements have been proposed (Elder, 1991), including using ASPN (an inductive modeling algorithm; see Elder, 1985; Elder and Brown, 1993) for modeling of separate regions. The polynomial models are everywhere differentiable (the chain rule of calculus applying to connectionist networks of polynomial nodes) so the estimated global minima can be found analytically by the algorithm.

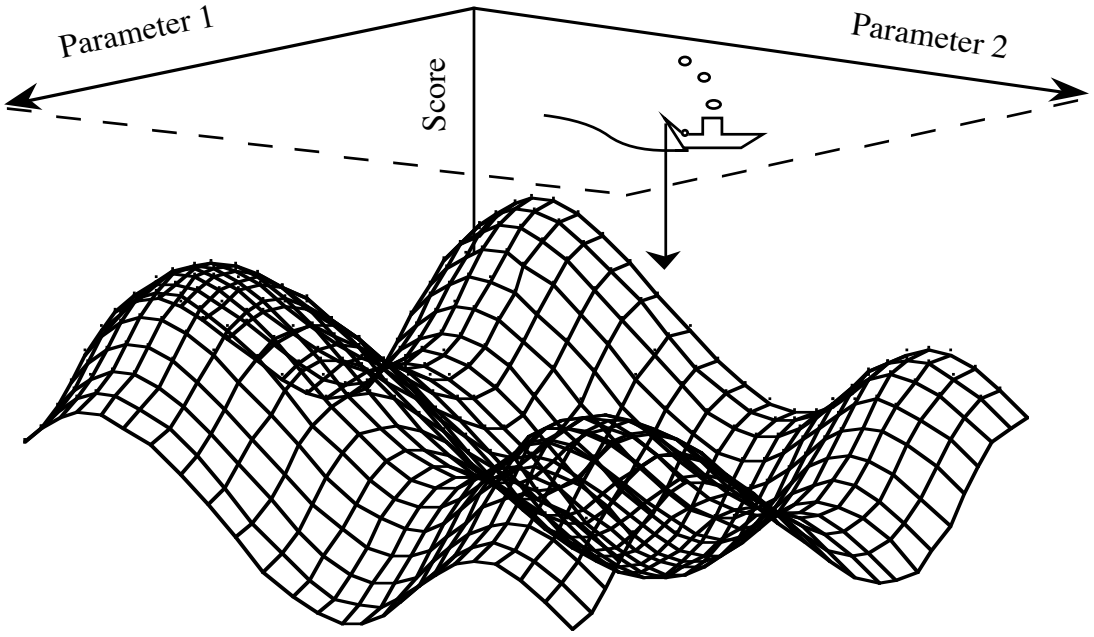


Figure 1: Global Optimization as Depth-Sounding (after Elder and Finn, 1991)

Models which might be even more locally responsive than "regional polynomials" could employ estimation techniques such as *kernels* (e.g., Parzen, 1962; Specht, 1967), *histograms* (e.g., Scott, 1985, 1992), *k-nearest neighbors*, *regression trees* (e.g., Brieman et. al., 1984; Chaudhuri et al., 1990), *splines* (e.g., Friedman, 1988), *radial basis functions* (RBFs) (e.g., Broomhead and Lowe, 1988), *wavelets* (e.g., Bock, 1992), or *locally weighted regressions* (e.g., Cleveland, 1979; Hastie and Tibshirani, 1986). (However, except for the many methods employing local quadratic polynomials, and a technique using RBFs, as discussed below, no such model-based search methods are known.)

More global estimation techniques, such as regression equations, or networks of *polynomials* (e.g., Ivakhnenko, 1968), "*artificial neurons*" (e.g., Werbos, 1974), or *logistic* elements (e.g., Cellucci and Hess, 1990), may not provide the local responsiveness which seems to be required of models used for optimization. That is, the local estimation techniques can be designed to *agree with the probe responses at the probe locations* -- an

important property when little or no noise is present. (On the other hand, the more global methods can provide better overall estimations in the early stages of the search, when there is not enough sample data to avoid the "curse of dimensionality".)

One form of *regional* model has recently been introduced (Skeppstedt, Ljung, and Millnert, 1992) which, though developed for control applications, could potentially be adapted for optimization. It uses a suite of models and a preliminary classification stage to identify which model is appropriate (an example of *structural* adaptation). (A similar function was implicitly performed by a single nonlinear network model, in (Elder and Barron, 1988; Barron et al., 1990) to robustly control a simulated reconfigurable fighter aircraft subject to damage of up to two control effectors simultaneously.)

The algorithm described in this work is based on the 1-dimensional *stochastic* model of Kushner (1964), which models $F(x)$ as a random walk in x. Torn and Zilinskas (1989) note that this was the first global optimization method to use a statistical model, and report its generally good fit to the response surface of many types of problems -- except in the neighborhood of a local minimum, when the uncertainty of F is not usually great enough to warrant a statistical model. Accordingly, Zilinskas (1976) introduced an improved 1-dimensional version which transitions to a quadratic local search when the results of a sequence of probes suggest the surface is locally convex. To indicate this transition in multiple dimensions, Mockus, Tiesis, and Zilinskas (1978) suggested noting regions which consistently return results lower than the expected value of the stochastic model.

Other statistical model-based methods include using *stationary stochastic functions* (Schagen, 1986), approximations of a full *Bayesian* approach (e.g., Mockus, 1989), and estimating *lower confidence bounds* (Cox and John, 1992). Additionally, there have been attempts to directly extend Kushner's algorithm to ^{d} (as described below).

In brief, the stationary functions of Schagen (1986) are assumed to have zero mean (to intercept the samples) and covariance relations such as

$$S_{ij} = \exp(-\theta \| x_i - x_j \|^2) \tag{10}$$

for some positive scale factor, θ. An *exploration* phase (identical in goals to the 1-step Kushner algorithm, below) is followed by a *location* phase, which samples at the lowest point of the model. However, as Torn and Zilinskas (1989) note, this location process may be unstable, as nothing prevents the expected value from being minimum in a region of negligible uncertainty. (Also, the aims of the location phase may instead be accomplished by a relaxation of the goal parameter during exploration, as discussed later.)

The Bayesian approaches of Mockus (e.g., 1988) have very strong theoretical properties, but the formal methods are unrealizable, and the approximations required to obtain usable algorithms are somewhat severe. For instance, one simplification of a Bayesian approach (Mockus, 1980) leads to a discontinuous response surface model. The mean function, $\mu(x)$, is piecewise constant (like a nearest neighbor, histogram, or regression tree model might provide), and the variance, $\sigma^2(x)$, increases linearly away from each sample (as with the Lipschitzian models).

The stochastic algorithm recently introduced by Cox and John (1992)[8] is called *Sequential Design for Optimization* (SDO), and employs an exponentially decaying radial basis function (RBF) to fit the surface with a *best linear unbiased predictor* (BLUP) (after Sacks, Schiller, and Welch, 1989). SDO evaluates the model on a grid of predictor locations, and probes the function at the best of those. In two-dimensional tests (Cox and John, 1992), SDO required significantly fewer probes to find the minimum (and to converge) than the best non-model-based algorithms studied.[9]

---

[8]From ideas proposed by J. Sacks and W. Welch.

[9]SDO did better on seven functions tested, and obtained comparable results on the eighth.

2.9 Summary

Though nothing like the vast literature on local techniques, the collection of search methods potentially appropriate for GROPE problems is yet large. Chapter 2 has surveyed the leading algorithms, using the six-class framework of Torn and Zilinskas (1989). The random (Section 2.4) and descent (2.6) methods behave locally (have "blinders on"), while the covering (2.3), clustering (2.5), and model-based (2.7-2.8) methods seek probe efficiency through constructing some sort of "big picture" of the emerging probe responses.

In our opinion, techniques which model the responses (Section 2.8) are the best for efficiently solving GROPE problems. They have also been tentatively recognized as the leading such methods recently (Cox and John, 1992) and in the past (e.g., Dixon and Szego, 1978).[10] Hence, the algorithm we describe in Chapter 3 is a multidimensional model-based method. However, it meets its design goal of probe efficiency (as empirically demonstrated in Chapter 5) only at the cost of considerable overhead calculations. In fact, in a taxonomy of techniques, the new GROPE algorithm (in its original form) is probably the most complex method with the most required auxiliary computations of all alternatives. Still, for the more complex GROPE applications with relatively expensive response functions, its probe efficiency can prove very valuable.

---

[10] It remains to be seen how much the new algorithm can extend the operating region of this class, which can be swamped by overhead as early as $d \geq 4$ or $N \geq 70$ (Mockus et al., 1978; Torn and Zilinskas, 1989).

3. The New Search Algorithm

The algorithm introducted here is a generalization of the 1-dimensional stochastic method introduced by Kushner (1962, 1964), and was inspired by the work toward that end of "the Louisville school" (Stuckman, 1988; Perttunen and Stuckman, 1990; Perttunen, 1991; Stuckman and Scannell, 1991). Recent refinements are believed to lead to a more theoretically consistent (and hence more efficient) ^{d} algorithm. The method is capable of discovering multiple extrema whether or not the function is differentiable and, as it builds on known results, can be paused and restarted with no waste of probing effort. A confidence measure is provided which approximates the probability that the final answer can be improved upon. This Section describes Kushner's 1-dimensional method, previous extensions, and the new algorithm for GROPE applications.

3.1 Kushner's 1-Dimensional Search

Kushner's optimization method for one dimension models the score surface, **y**, as a *random walk* in **x**; that is, as a *Brownian motion* or *Weiner* process, where the y value at a point x is Gaussianly distributed with respect to neighboring points. An example "drunkard's walk" is shown in Figure 2; the path is that of one moving forward at a constant rate, but staggering random amounts to each side. The distribution of stagger steps is Gaussian, and for a time series with discrete steps, y_{t+1} can be shown to be $N(y_t, \square^2)$. That is, the mean value for the next point is the current point; no information about the direction of the step is available. Knowledge of how the curve got to y_t is of no use; the distribution is *memoryless,* or *Markovian*, and the only values affecting the estimator of an unknown point are those of its nearest neighbors in x.

The random walk is descriptive of many processes, from the Brownian motion of particles suspended in a liquid to the price history of the "S&P 500" stock market index. As a model, it has the advantage of being *fractal* or *locally rough* (it is nowhere differentiable)

but *regionally smooth*. Thus, it is possible, though unlikely, for a large jump in *y* to occur between points close in *x* but, most near differences will be small, and the surface is broadly "rolling" -- a representation capable of fitting many practically occurring functions.

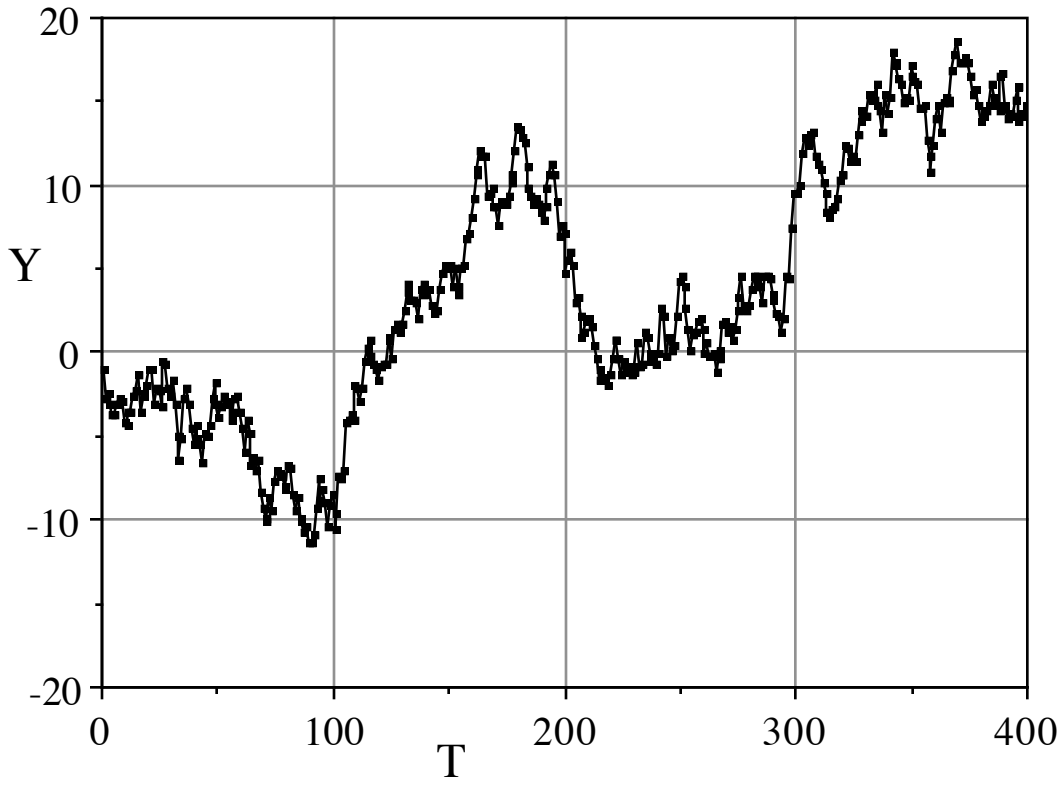


Figure 2: An Example Random Walk: $Y = \sum X(t)$, $X=N(0,1)$

The random walk assumption is weak, allowing the algorithm to be effective on rough response surfaces as well as the smooth ones addressable by methods with tight continuity or differentiability assumptions. Though some efficiency is lost on smooth functions by not taking advantage of their restrictions, the algorithm is competitive on them with other methods (as shown in Section 5), yet also has the ability to handle rough functions.

A further practical advantage of the representation is its tractability. Let $y_i = F(x_i)$. In the case of no noise, the Markovian property implies that the conditional expected value of *y* at a position *x* between two known points *a* and *b*, is the linear interpolant

$$\mu(x \mid \mathbf{x},\mathbf{y}) = y_a + p(y_b - y_a) \tag{11}$$

where the proportion p is $\frac{x-x_a}{x_b-x_a}$, as shown in Figure 3a. Also, the variance conditioned on all previous results is a quadratic function of the distance from the interpolating bounds

$$\sigma^2(x \mid \mathbf{x}) = cp(1 - p)(x_b - x_a) \tag{12}$$

for some slope factor, c (the mean squared variation in y as x changes). (Note that σ^2 has no other dependence on the y values.) This variance grows linearly with distance when only one neighbor is known (while the mean remains constant at the edges), as shown in Figure 3b. When noise is present (i.e., probes at the same location can return different values), the representation is only slightly adjusted (Kushner, 1962): $\mu(x)$ does not go through the samples exactly, but *shrinks* toward neighboring samples, and $\sigma^2(x)$ is positive, not zero, at the probe locations.

Kushner (1964) solved for the probe location most likely to exceed the current best value by a fixed amount (and suggested this magnitude could change with time, reminiscent of the "temperature scheduling" strategy which directs simulated annealing. A slightly different perspective (Stuckman and Scannell, 1991) is to seek the point most likely to exceed a given *result goal*, y_g; i.e., to find the $\mathbf{x}$ maximizing

$$\Pr[y > y_g \mid \mathbf{x},\mathbf{y}] = 1 - \Phi\left[\frac{y_g - \mu(x|\mathbf{x},\mathbf{y})}{\sigma(x|\mathbf{x})}\right] \tag{13}$$

This is depicted in Figure 4 for a one-dimensional line segment. Points close to x_b have the advantage (in putting probability mass across y_g) of starting closer; yet, locations in the middle of the segment step farther. Thereby, the conflicting aims of exploration and exploitation are balanced. The goal-exceeding objective is also appealing when a natural bound is available, whether from known or theoretical limits (e.g., zero error or, a competitor's results!). When a value is not available however, the algorithm can employ the

usual "carrot on a stick" approach, and strive to beat the current best result (by a possibly dwindling amount).

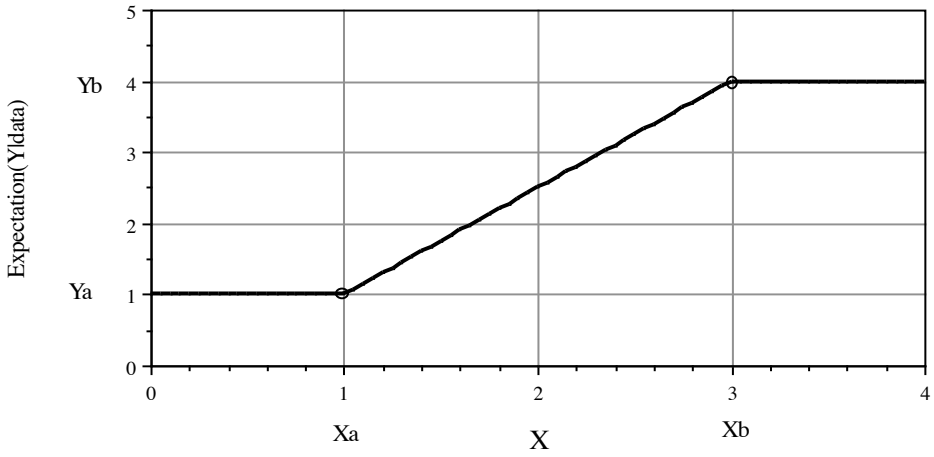


Figure 3a. Expectation of Y Conditioned on Y_a and Y_b

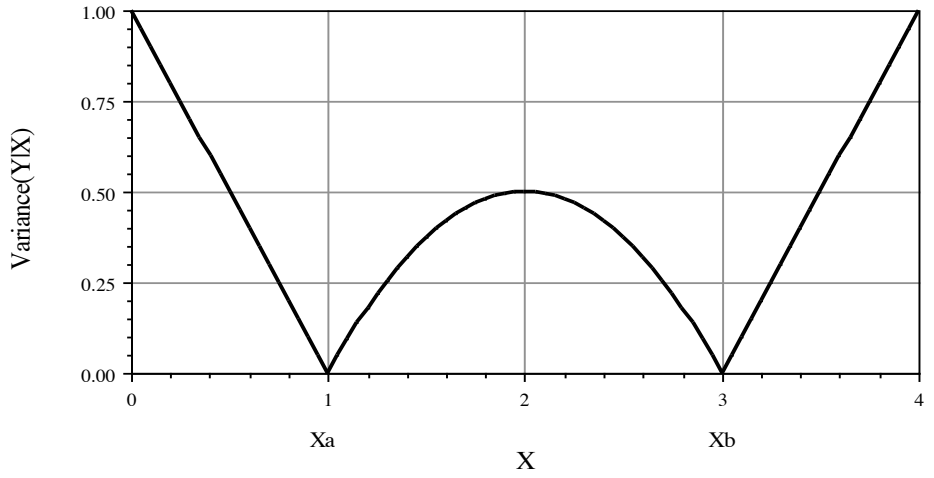


Figure 3b. Variance of Y Conditioned on Known Values at X_a and X_b

As □ is monotonic, we may maximize (13) by *minimizing* its argument, (or square, using (12)). Substituting, this translates to finding the proportion p which minimizes

$$D^2(p) = \frac{[y_g - (y_a + p(y_b - y_a))]^2}{cp(1-p)(x_b - x_a)} \qquad (14)$$

Solving $\frac{\partial D^2}{\partial p} = 0$ reveals that the optimal proportion depends only on the relative distance of the end points to the goal

$$p^* = \frac{\Delta a}{\Delta a + \Delta b} \qquad \text{where } \Delta a = y_g - y_a,\ \Delta b = y_g - y_b. \tag{15}$$

(Note that the slope parameter, c, has no influence on p^*, and may be dropped.) $D^2(p^*)$ is monotonic with the segment's maximum conditional probability of containing a location exceeding the goal.

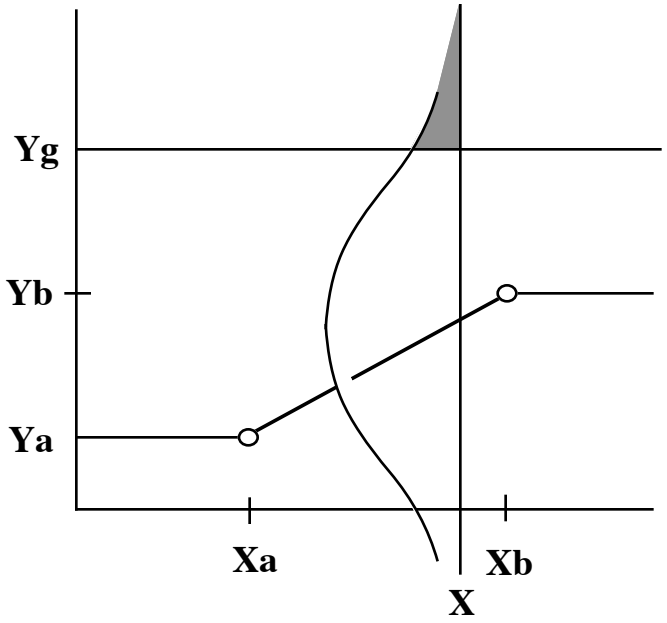


Figure 4: Find $X \in [X_a, X_b]$ to Maximize Pr[Y > Y_g]

The 1-dimensional algorithm can be summarized:

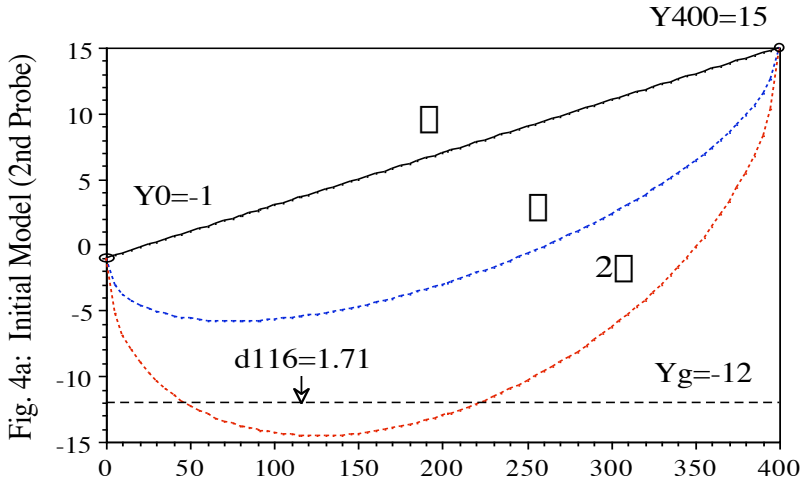
1) Probe the two bounds of the search space.[11]
2) Calculate the best sampling location, p^*, for the line segment, and insert that location, x^*, in an ordered list according to its distance estimate, $D^2(p^*)$.
3) Remove the top member of the list and probe at x, breaking the line segment (a,b) into (a,x) and (x,b).
4) Repeat steps 2 and 3 (with two segments for step 2) until the goal is exceeded, resources are exhausted, or the probability of better results is acceptably low.

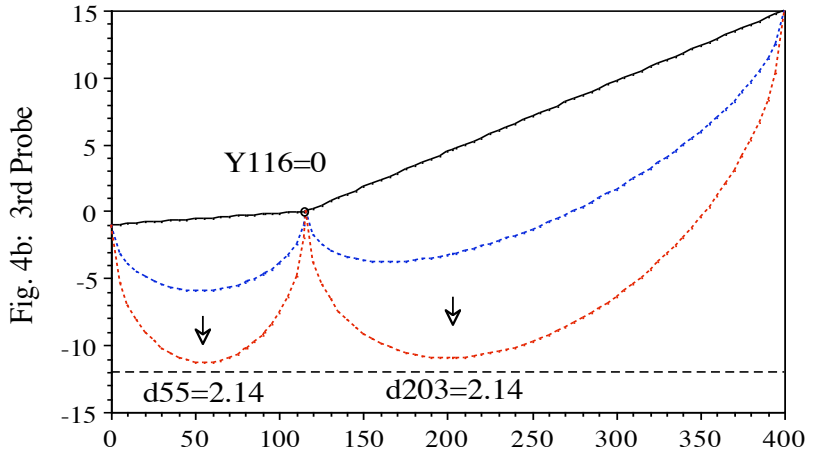
Figures 5a through 5e illustrate several steps of the Kushner algorithm for the data of Figure 2. (The piecewise linear model, shown with one- and two-σ lower bands, is updated with each probe. Each result adds a pair of equidistant locations, identified with arrows, to the set of candidate probes. Leading candidates are depicted with open arrows. In the example depicted, ties were broken at random. Note that the goal of $Y \leq -12$ can be reached on the next (7^{th}) probe, at $X = 72$.)

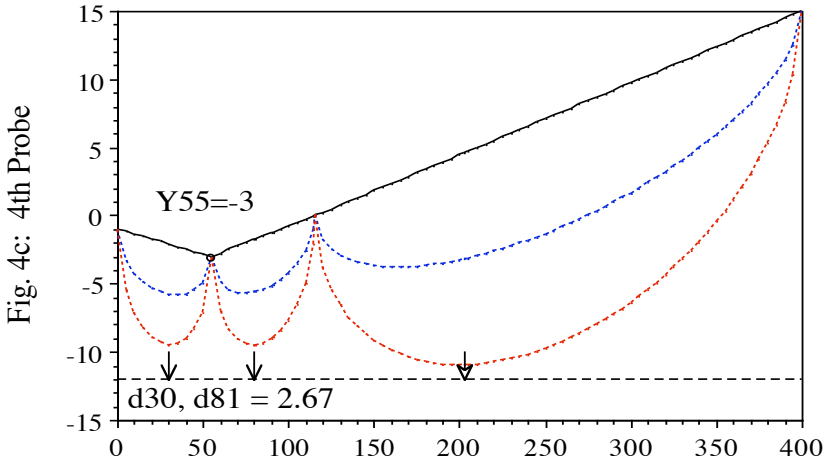
After each probe, the list of candidate probes increments, as one segment is removed and two take its place. Unless the goal changes, in which case the optimal location for each segment must be reevaluated, only the p^* locations for the two new segments must be computed. (In the Appendix, it is shown that their two distances are identical; see e.g., Figure 5b. In that example, ties were broken randomly.) When results are far from the goal (at the beginning of the search), the variance component of (14) dominates and locations relatively midway between known probes are preferred. When the best probes score near the goal, further probes in their vicinities are called for. This mechanism performs the tradeoff between the conflicting search aims of 1) homing in on the extreme of a promising area, and 2) sampling unknown regions.

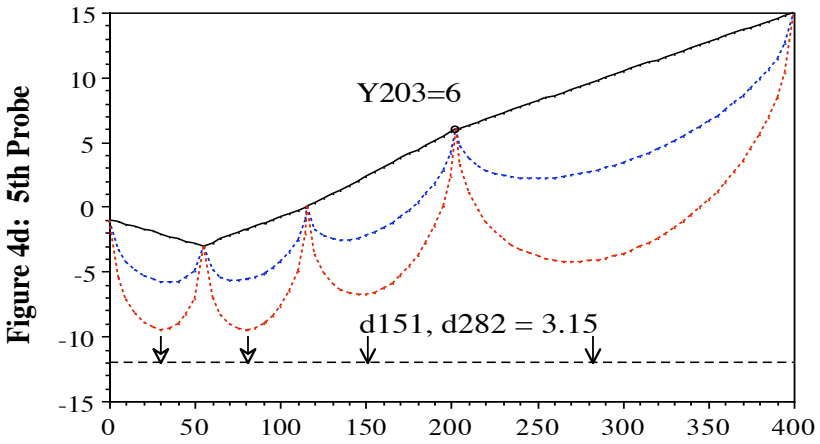
---

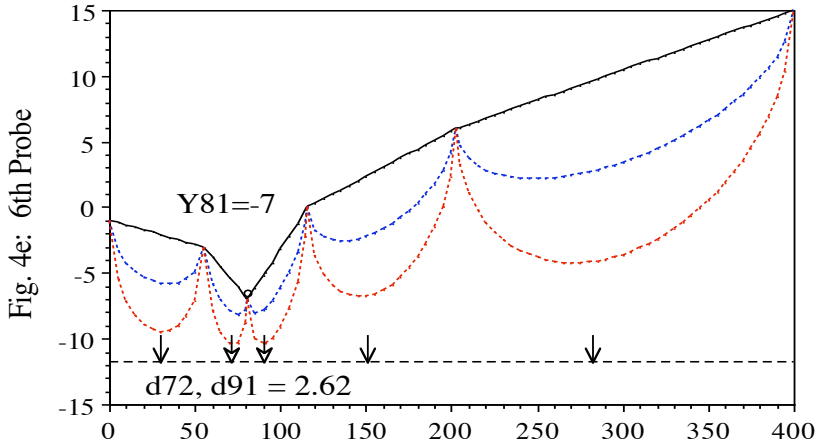
[11] The initializing probes can be inside the bounds (for three line segments initially). Still, probes may later be sought on the edges of the legal space anyway, since σ^2 grows rapidly beyond the outermost probe.


Y400=15
Y0=-1
d116=1.71
Yg=-12
Fig. 4a: Initial Model (2nd Probe)


Y116=0
d55=2.14
d203=2.14
Fig. 4b: 3rd Probe


Y55=-3
d30, d81 = 2.67
Fig. 4c: 4th Probe





3.2 Extension to Multiple Dimensions

The key difficulty in expanding Kushner's algorithm from 1 to d -- and perhaps the reason the method saw little use for a generation -- is the extension of the random walk model into a random *field* (for which there are even competing theoretical definitions in the literature; e.g., Adler, 1981).

The *multi-univariate* method (Stuckman, 1988) avoids the issue of random fields, and employs the one-dimensional algorithm along the line segments connecting all pairs of probes (or a subset of *K*-nearest neighbors for each probe, where *K* temporarily jumps an

order of magnitude when the probe is the current best). However, such a procedure can ignore a probe intermediate to another pair, as depicted in Figure 6. More importantly, the method is silent about function values everywhere except on the connecting line segments -- a zero measure subset of the search region. In a similar manner, the modified Kushner's algorithm of Zilinskas (1976) has been employed in multiple dimensions using sequences of line searches. Results on standard test problems have been generally discouraging (Dixon and Szego, 1978), though Torn and Zilinskas (1989) define (what they believe are more realistic) situations when such a line technique would be useful.

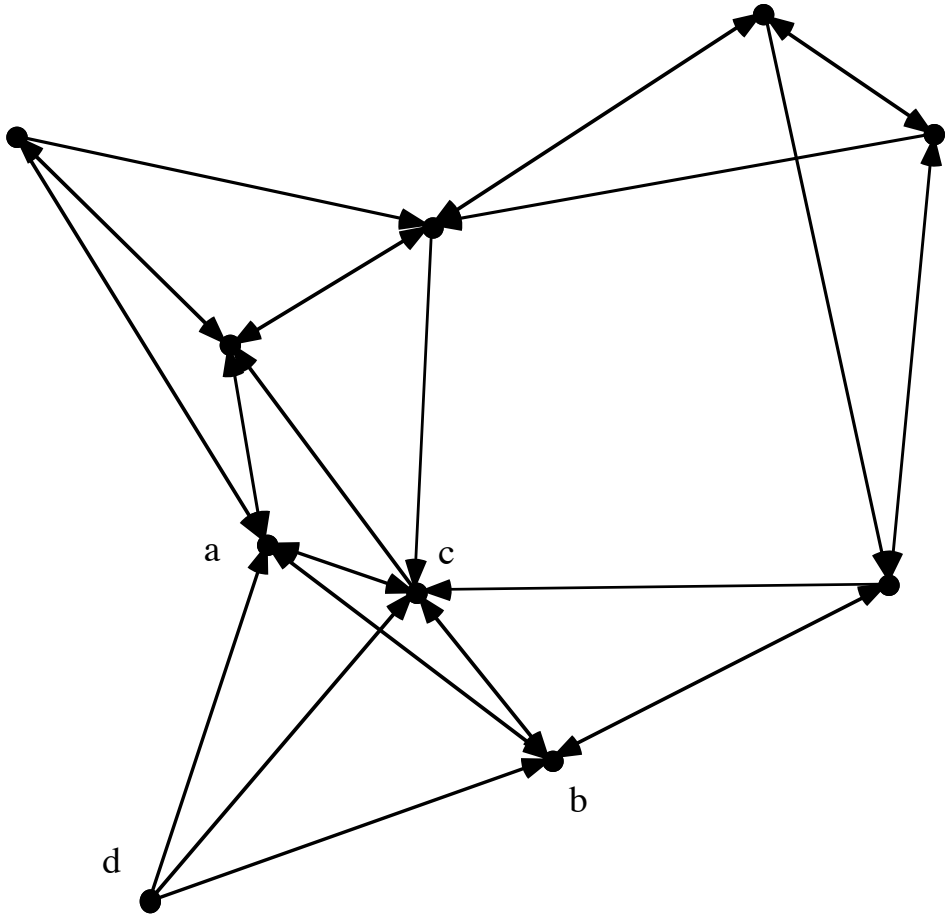


Figure 6. Connection of 3-Nearest Neighbors for Set of Example Probes. Note that probe *c* would have no influence on estimation along segment *ab*, and that nearest neighbors are not symmetric.

To instead *cover* the search space, the region within the *convex hull* of the probes can be *tessellated* (divided into space-filling disjoint regions) into a set of *simplices*. In ^{d}, a simplex is a convex polyhedron composed of d+1 vertices (i.e., a triangle in two dimensions; a tetrahedron, in three). If a *simplex subdivision* approach is employed (e.g., Groch et al., 1985), a new probe divides its surrounding simplex into d+1 smaller simplices (defined by the new point and each *face* of the old simplex), leaving all other regions intact. It would be better, however, to update the entire tessellation in a manner maintaining some optimality property, such as *local equi-angularity* (Lawson, 1972) in which small angles in triangles are avoided. The unique set of connections with this property in the plane is the *Delaunay* triangulation (Sibson, 1978). (However, in three and higher dimensions, this triangulation does not necessarily maximize the minimum *solid angle* (Joe, 1989).)

The Delaunay triangulation (shown in Figure 7 for the points of Figure 6) is the dual of the *Voronoi* (or Dirichlet or Thiessen) tessellation, wherein regions are partitioned according to the nearest neighbor rule (as shown in Figure 8). That is, all the points within a region are closer (by L_2) to the same known probe than they are to any other. Another property of the triangulation has long been known for low dimension (Miles, 1970), but only recently proven in general (Rajan, 1991): the *circumscribing sphere* of each simplex is empty. That is, the only triangulation in which the sphere intersecting the vertices of a given simplex contains no other point, is the Delaunay (a construction illustrated in Figure 9). (Rajan further proved that the maximum radius of the smallest of such spheres is less than that for any other triangulation -- allowing the Delaunay triangulation to be formulated as a solution to a continuum optimization problem.) The earliest high-dimensional algorithms based on the empty circumsphere property are the virtually simultaneous versions of Bowyer (1981) and Watson (1981). The design of efficient programs, especially for the incremental case -- in which a new probe is continually introduced to an existing triangulation, remains a

subject of considerable current research (e.g., Joe, 1991; Edelsbrunner and Shah, 1992; and Hearne and Wegman, 1992).

The optimization algorithm of Perttunen (1991) employs Delaunay triangulation to tessellate the search space, but scores each candidate simplex with a heuristic, nonparametric metric: the product of the *ranks* of the vertices divided by its content, or "hypervolume" (Perttunen and Stuckman, 1990). The next probe is taken within the winning simplex, at the weighted average location of its vertices (where the weights are the inverse relative ranks of the probe scores). They show that Delaunay triangulation improves the ranking method, causing the scatter plot of search points to better correspond to the contour diagram of each objective function tested. However, use of the ranking, coverage, and weighting heuristics lead to a technique having little in common with Kushner's stochastic algorithm.

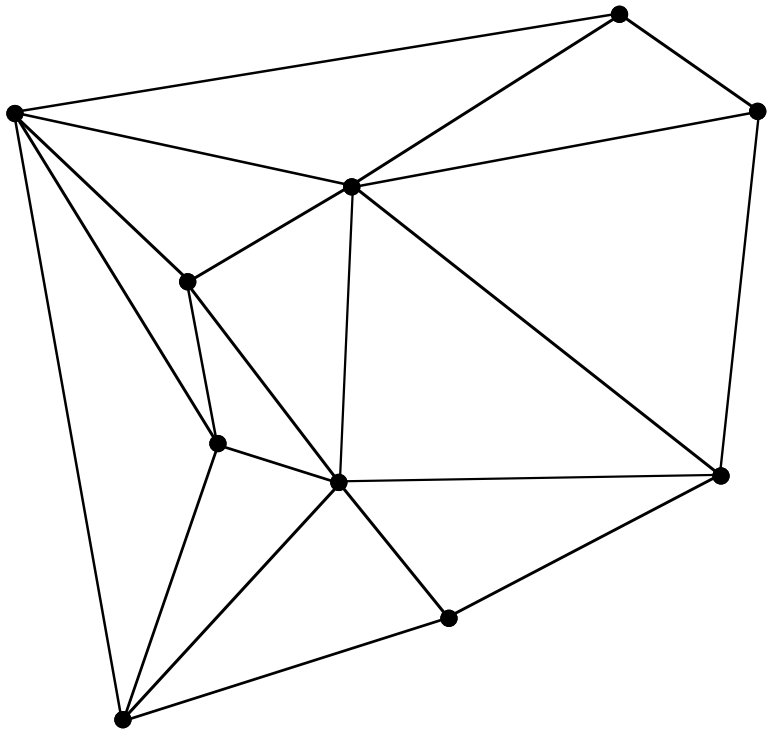


Figure 7: Delaunay Triangulation (of Points from Figure 6)

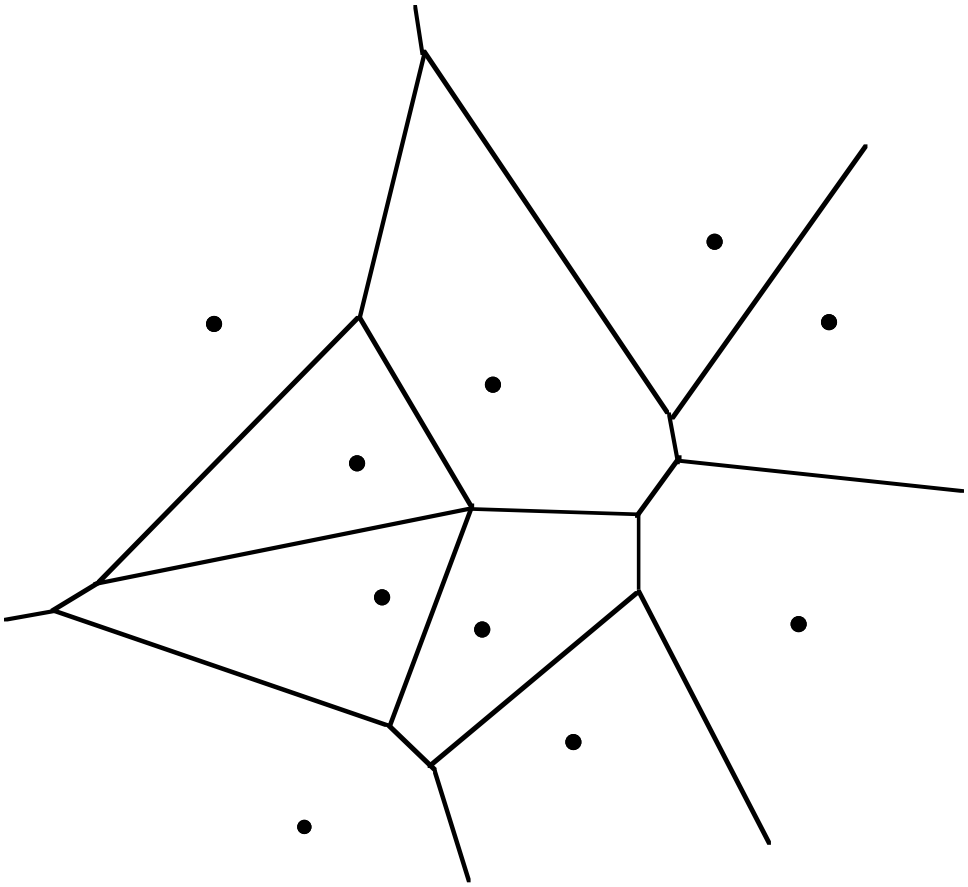


Figure 8: Voronoi Tessellation (of Points of Figure 6)

Building on the ideas of tessellation and goal-direction however, a more straightforward generalization is possible: use a linear interpolation of the response values at the Delaunay vertices to define the conditional expected values inside a simplex, and a quadratic polynomial for the conditional variance, constrained to agree with Kushner's variance curve along each 1-dimensional simplex edge. The expectation is thus a piecewise planar surface, resembling facets of a gem -- a representation similar to that of the two-dimensional *marching cubes* algorithm (Lorensen and Cline, 1987), or of *hinging hyperplane* models (Breiman, 1991). For example, the 2-dimensional Delaunay triangulation of probes in Figure 10, leads to the interpolation surface of Figure 11.

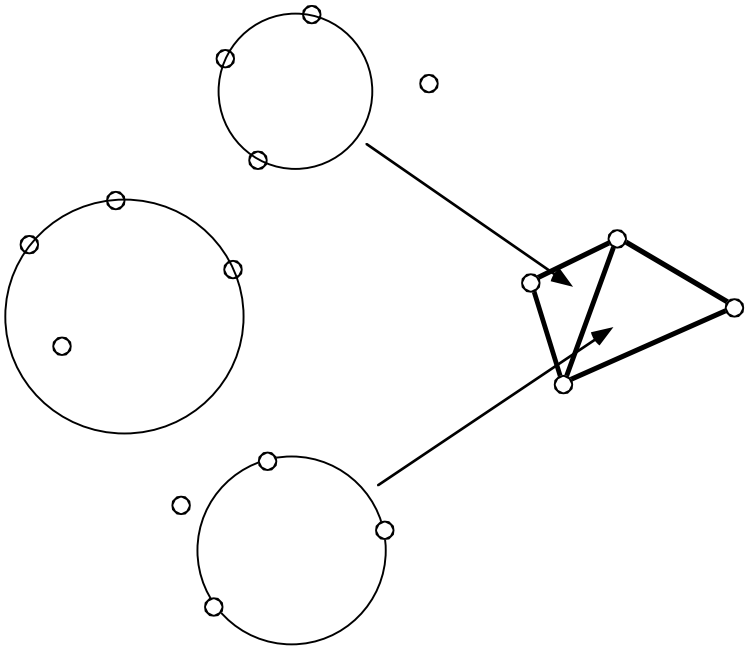


Figure 9: Delaunay Triangulation of Four Points
According to the Empty Circumsphere Rule

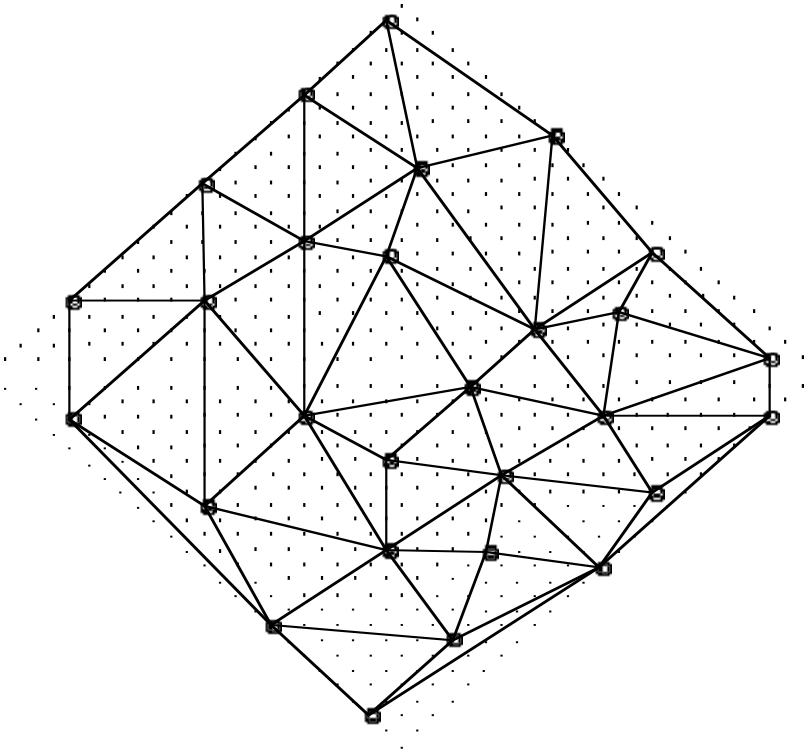


Figure 10: Delaunay Triangulation of 28 Probes
(from 25x26 Grid of Potential Sites)

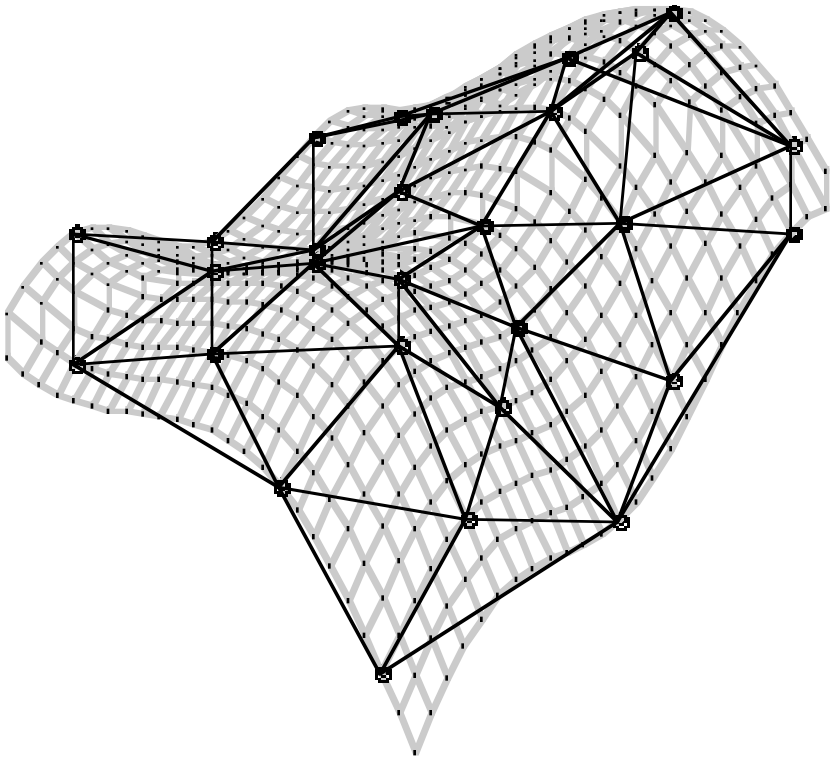


Figure 11: Triangular Facets Interpolate Function Surface

The relative variance "canopy" arches over the simplex as shown in Figure 12, from lows at locations with known values, to an interior peak far from the vertices. This variance can be defined by the (unique) complete quadratic polynomial in d variables which conforms to the univariate equations along the $\binom{d+1}{2}$ edges of the simplex (and is undefined outside these bounds). These variance constraints are imposed since a hyperplane defined by the vertex values is used for the expectation. The variance can be viewed as a measure of uncertainty about the mean, so their methods of estimation must be linked. Along an edge, only the pair of connected vertices affect the conditional mean value[12] (as with Kushner's 1-dimensional method); therefore, the edge constraints on variance are *necessary* for this generalization of the algorithm.

---

[12]This 1-dimensional property can have the side-effect of ignoring the nearest known probe, as can happen (for a thin simplex near the convex hull) when estimating the values of an edge segment near a third vertex. Thus, the algorithm actively avoids the creation of thin boundary simplices (4.2).

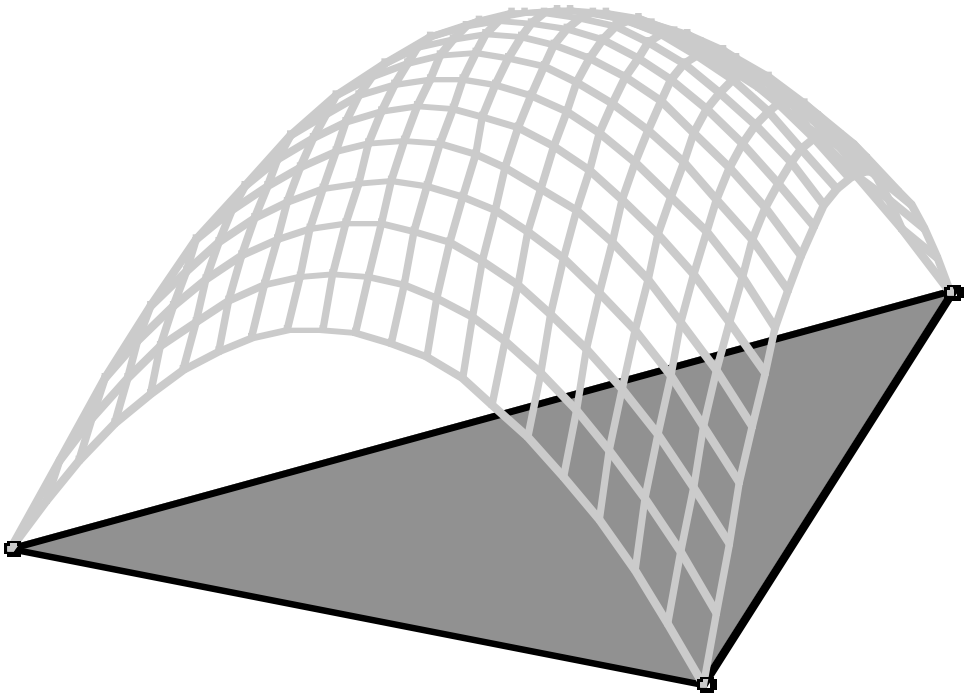


Figure 12: Example Variance Canopy

The edge constraints are also *sufficient*. There are $\binom{d+2}{2}$ parameters in a second-order polynomial in d variables. (In general, the complete polynomial with maximum power p has $\binom{d+p}{p}$ terms). Each of the $\binom{d+1}{2}$ edges of the simplex contributes one constraint -- say, the value of the variance at the midpoint of the segment (which, for a given c, is defined by its length (2)). The remaining d+1 constraints are provided by the vertices of the simplex, for which the variance is a minimum fixed value (zero for noiseless probing). Since none of the edges are collinear (as demonstrated by the existence of a circumsphere for the simplex), the exact match of constraints and degrees of freedom means the polynomial solution will be unique and have zero error.[13]

---

[13]Thin triangles on the convex hull can lead to nearly collinear edges. It had been proposed (Elder, 1992) that robust regression techniques such as singular value decomposition be employed to remove near-singularities, but the preventative method of *attractive boundaries* (4.2) was found to be, well, more attractive.

The locations and scores of the d+1 probes of each simplex thus define the equations for the linear expectation, $\mu(\mathbf{x})$, and quadratic variance, $\sigma^2(\mathbf{x})$, of its interior (which may be obtained via linear regression). In one dimension, the optimal interior location, x^*, for each simplex is known analytically (15). This can also be shown to be the case for two dimensions, but the solution is surprisingly complex. For multidimensional applications, an easier approach is to perform an internal search of the function to be minimized

$$D^2(\boldsymbol{x}) = \frac{(y_g - \mu(\boldsymbol{x}))^2}{\sigma^2(\boldsymbol{x})} \tag{16}$$

As shown for Figure 13, this squared distance function is positive, smooth, analytical, and (believed to be) unimodal -- allowing any of several local minimizers to be employed.[14]

[For some reason, there were technical difficulties translating this Figure to .pdf. In brief, it looks like a net stretched over a triangular space, with the surface peaking at the vertices and relatively flat in the interior.]

Figure 13: Example $D^2(\underline{x})$ Surface and its Minimum

[14]However, $D^2(x)$ is not defined outside the simplex, and explodes at the vertices, so care must be taken at the boundaries. This is done implicitely through *compositional mapping* (4.3.2)

3.3 The Search Algorithm

The search can be initialized by probing at least d+1 points from the convex hull defining the space, **A**, or by absorbing previous results.[15] (As all probes contribute to the model locally, the program can pick up where a prior run left off.) Then, until the goal, Y_g, is reached, resources run out, or the probability of improvement (e.g., equation 7) is sufficiently slight, the algorithm continually probes at the current most promising location and updates the model with the result.

The basic steps of the algorithm (discussed in more detail in the next Chapter) are:

1. Probe at the boundary locations and construct the initial Delaunay Triangulation (DT) (4.1.2). Probe also at the mean of **A** (4.1.1).
2. Update the DT (4.1.3), removing representatives of obsolete simplices from the ordered list of candidate probe locations.
3. For each new simplex j:
   a) Solve for the expectation plane, $\eta_j(\mathbf{x})$, given the vertices (3.2).
   b) Solve for the quadratic variance canopy, $\sigma^2_j(\mathbf{x})$, given the vertices and edges (3.2).
   c) Find the best probe location, $\mathbf{x}^*_j$, for the simplex by minimizing (16), the squared, standardized distance, $D^2_j(\mathbf{x})$, from $\eta_j(\mathbf{x})$ to the goal, Y_g (4.3).
   d) Insert this probe location into a candidate list ordered by $D^2_j(\mathbf{x})$.
4. If Y_g has changed, update old simplices and their candidates (4.4).
5. If not finished, probe at the best location (the head of the list), and continue at Step 2.

---

[15] Currently, **A** is assumed to be a d-rectangle; i.e., univariate bounds are employed (see 4.1.1).

Steps 2 and 3c, the re-triangulation and the internal search of new simplices, are most affected by the problem dimension *d*, (though are virtually independent of the number of known probes, *N*. The added overhead is rather great (compared even to some other model-based searches). However, for low dimensions, when probe computations are not trivial, that time should be more than compensated for by the algorithm's judicious choice of locations.[16]

3.4 Convergence

As the variance argument prevents probes from being repeated, a straightforward proof of convergence is possible.

Theorem 2

The GROPE Algorithm converges with probability 1 to a δ-neighborhood of $\underline{x}^*$.

The Voronoi Tessellation covers R^d; its dual the Delaunay Triangulation, defined within **A**, thus covers **A**. If Y_g is always $< Y_{min}$, all interior locations, $\underline{x} \in \mathbf{A}$, will have positive distances, $D^2(\underline{x})$. This distance, to be minimized within each simplex, is inversely proportional to $\sigma^2(\underline{x})$, which is zero at known locations, $\underline{x}_i$, $1 \le i \le N$. Any valid probe thus reduces the overall variance of the model; i.e., provides new information. No location is re-visited. As new probes reduce the average (and maximal) content of the set of covering simplices, and as no candidate locations are discarded, the result follows. That is, the algorithm asymptotically converges with probability 1 to a δ-neighborhood of $\underline{x}^*$.

---

[16] For example, in (Elder and Barron, 1988) each probe for a control application consisted in running a full computer simulation with a new set of parameters. Such a task can easily take minutes per probe on a workstation (and engender, in the early morning hours, a visceral distaste for senseless search methods!).

4. Algorithm Implementation

This Chapter discusses issues key to implementing the general-dimensional search method discussed in Chapter 3, and describes modifications that improve its practical use.

4.1 Delaunay Triangulation

Delaunay Triangulation (DT) is the dual of Voronoi Tessellation, wherein the search space is divided into regions within which any point is closer (by the Euclidean metric) to a given known probe than to any other. The DT connects sets of d+1 probes that share a Voronoi boundary and the center of the circumscribing sphere for each such simplex is a Voronoi focal point, where d boundaries meet. In the response surface model maintained and updated by the search algorithm, each simplex is an "island", unaffected by probe results outside its bounds. This is a consequence of the assumed random field model and the "linear dominance" generalization of the memoryless property of the random walk. The resulting model is thus very flexible and responsive to local variations and capable of representing multi-modal and irregular response patterns. On the other hand, little advantage will be taken of response surfaces which are, in fact, highly structured (e.g., quadratic), though the method will be shown to remain competitive with other global techniques even on such problems (Chapter 5).

The central use of DT is thus an important distinctive of the algorithm, providing it both strengths and potential weaknesses. One area of concern is the burden of overhead it imposes. In worst case, updating the DT can be an $O(N^{\lceil (d+1)/2 \rceil})$ process (Seidel, 1982) for N probes, though it has recently been shown to have linear expected complexity, $O(N)$ (Dwyer, 1989). In most cases, use of only the nearest surrounding simplex of probes for estimation at a point is quite reasonable; however, the underlying assumptions can be somewhat strained at the bounds of the search space (where simplices can be far from equilateral), requiring special attention (4.2).

4.1.1 Rectangular Bounds

For simplicity, the search space, **A** is assumed to be rectangular:

$$\mathbf{A}: \{ \underline{x}: x_j \in [b_{0j}, b_{1j}],\ 1 \le j \le d \} \tag{17}$$

for lower and upper bound vectors, $\underline{b}_0$ and $\underline{b}_1$. However, the set of probes located at the corners of the *d*-rectangle are *degenerate*, as the triangulation is not unique; in fact, even the number of triangles[17] for a given *d* may vary. (A 3-dimensional rectangle, for example, can be composed of either 5 or 6 tetrahedrons.) This causes problems for some Delaunay Triangulation (DT) algorithms; yet conversely, can be viewed as a positive property, as any valid triangulation will be a Delaunay one. The particular triangulation initializing the space is particularly fragile however, as the first interior probe (number $2^d + 1$) will "shatter" all initial triangles and be a vertex in each of the 2^d new ones.

On the other hand, in some cases the initial triangulation -- essentially an arbitrary choice among a set of possibilities -- can have an undue effect on the early steps of the algorithm. For example, if responses at diagonal corners of a 2-dimensional space are very high relative to the others, the two alternative triangulations lead to the rather different estimated surfaces of Figure 14. For all but very distant goals, the initial candidate locations will be in the "valley" of Figure 14a and on the "foothills" of Figure 14b.

With subsequent probes, the "nondeterministic" effects of the initial DT choice are reduced; however, to reduce them further, the first interior probe can be pre-selected to be at the mean of the space (also the center of all the simplex circumspheres). As the DT is completely overhauled by this probe, candidate locations and centroids, etc. do not need to be calculated in the early initialization phase (described next), saving some time.

---

[17] For simplicity, "triangles" will often be used to refer to simplices, regardless of dimension.

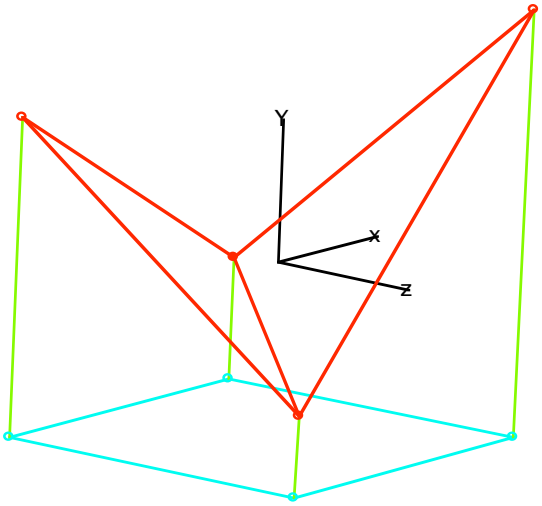


Figure 14a: "Valley" Triangulation of 2-Rectangle

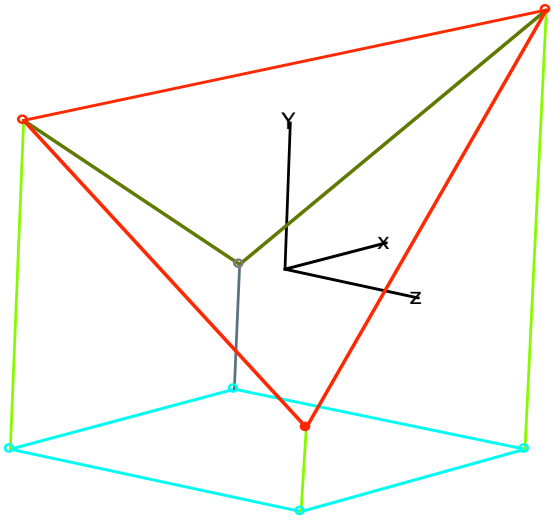


Figure 14b: "Foothill" Triangulation of 2-Rectangle

4.1.2 Initialization

A simple DT scheme was suggested by Joe (1993) wherein a chosen point (vertex probe) is connected to each opposing *face* ((*d*-1)-dimensional space), the set of *d* sub-spaces covering **A**. Each sub-space is then recursively triangulated into simplices with one fewer point, and connected to the original point. As a consequence, the original point (and its opposite, it turns out) are members of all *d*! initial triangles.

Based on that idea, a slightly simpler recursive scheme was developed wherein a list of d bound values is updated and made into a triangle whenever full. Corners are identified by an integer with d bits (0 to $2^d - 1$), where bit j is 1 if x_j is on the high bound, 0 if low. For example, corner 26 (= 11010) is low on x_1 and x_3, and high on x_2, x_4, and x_5. To initialize the triangulation of the corner probes, call `slice_up( 0, 0 )`:

```
slice_up( constr, pos ):
    list(pos) = constr
    if (pos equals d)  make_triangle with list
    otherwise, for every bit j in constr that is zero:
        slice_up( constr + j_bit, pos + 1 )                    (18)
```

Given a partial list of probe components and the current constraint (an integer of d bits) the routine adds a corner to the list and, for the next position, calls itself once for every zero in the constraint (changing that zero to a one). The base case occurs when the list is full. The resulting triangles are those that would be formed under the Joe (1993) scheme.

4.1.3 Updating the Delaunay Triangulation

The DT is updated according to a (corrected and generalized) 2-dimensional algorithm due to Watson (1981), which hinges on the empty circumsphere rule. When a new probe *shatters* a triangle (is inside its circumsphere) that triangle is dismantled and its edges added to a list. When all triangles have been checked, duplicate edges in the list are both removed and new triangles are formed from the new probe and each of the remaining edges listed.

As Rajan (1991) recently proved that the empty circumsphere property holds for general-dimensional DTs, the Watson algorithm was examined (with simplex faces replacing edges) and found to work except when the probe is on a boundary. Then, even the original 2-dimensional version fails as the boundary face is incorrectly listed. (Probes on an internal simplex face are, by definition, on two, so the face is correctly unlisted by the duplicate

provision.) The procedure is corrected by not listing any face intersected by the new probe (of which there may be up to d-1 in the noiseless case). This is important, as boundary probes occur somewhat frequently, especially if they are actively sought after to improve the global properties of the triangulation, as described below (4.2).

4.1.4 Circumcenters

To maintain the DT, the circumscribing sphere for each simplex must be found. As the DT is the dual of the Voronoi polytope, the sphere's center is located at the intersection of all the nearest-neighbor boundary planes separating each pair of points in the simplex. That is, the d center coordinates, $\underline{c}$, can be found by solving a set of simultaneous equations for any d of the $\binom{d+1}{2}$ planes (of dimension d-1) equidistant from a pair of vertices of the simplex.

Making equal the Euclidean distance from points $\underline{x}$ to vertices $\underline{p}$ and $\underline{q}$

$$\sum_{j=1}^{d} (p_j - x_j)^2 = \sum_{j=1}^{d} (q_j - x_j)^2 \tag{19}$$

leads to the linear equation $\underline{a}^T\underline{x} = b$, where

$$a_j = p_j - q_j, \quad j = 1, d \quad \text{and}$$

$$b = \frac{1}{2} \sum_{j=1}^{d} (p_j^2 - x_j^2) \tag{20}$$

to define the plane.[18] At least d of these planes can then be solved for simultaneously using a linear least squares procedure to find the unique point, $\underline{x}_c$, defining the center. This location and squared radius, r^2 are retained by each simplex for use in maintaining the DT.

---

[18] Thanks are due Bob Ferguson for (19), clarifying the derivation of (20).

4.2 Boundary Attraction

The Delaunay triangulation is that which maximizes the minimum angle in two dimensions. In general, it leads to relatively equi-angular simplices, except when a probe is very near, but not on, a search boundary. The boundary remains linearly non-dominated by other points, so is retained as a face, resulting in very "skinny" boundary simplices, with low interior-to-surface *content* ratios.

In practice, the nearly collinear simplex edges can lead to numerical instabilities but, more importantly perhaps, the unchanged existence of the large boundary face means little illumination of the larger picture is likely to be made by the probe. In particular, the relatively large variance of the boundary (due to the distance between endpoints), results in a new candidate probe much like the last, leading to ever-smaller steps towards the boundary (like the apparent paradox of Xenon's arrow approaching, but never striking, a target.) The resulting clusters of mid-boundary probes are apparent on each edge of the space shown in the example of Figure 15a.

A solution is to identify when the leading probe would be "very close" to a limiting condition (a boundary) and to nudge it there. The threshold distance, δ, used to determine "close" is a small percentage (e.g., 1%) of the span of a boundary, $b_{1j} - b_{0j}$. This change in the example leads to the slightly improved distribution of probes shown in Figure 15b.

It makes sense to nudge further when the simplex itself spans a large space so, alternatively, the longest chord in the simplex, or the radius of its circumsphere could be employed to adjust δ. Then probes would be pulled greater distances early in the search (when the triangles are large) and fewer clusters of probes on the boundary would occur.

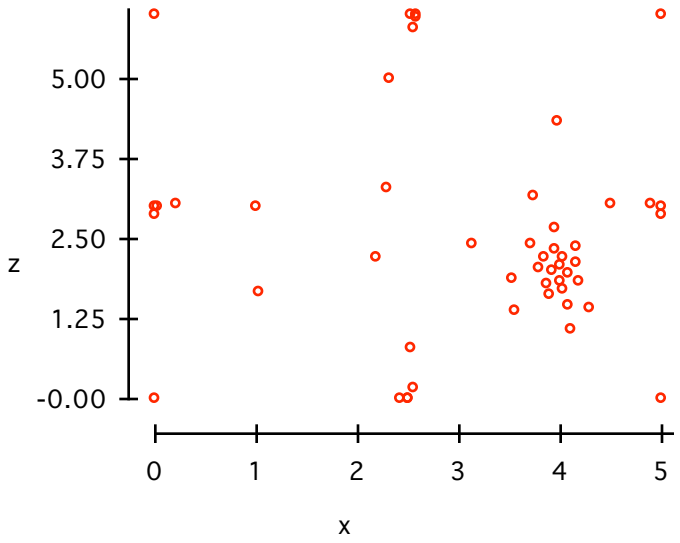


Figure 15a: Example Probe Distribution

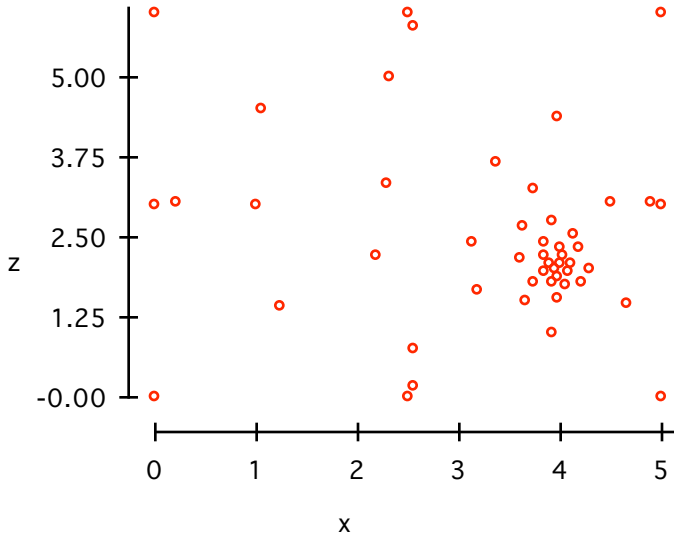


Figure 15b: Example Probe Distribution with Slight Boundary Attraction

4.3 Internal Search

The function (16) to be minimized within the domain of each simplex estimates the standardized distance from interior points to the current response goal, Y_g. It consists of a monotonic[19] convex quadratic numerator and a non-monotonic concave quadratic

[19] along any 1-dimensional projection.

denominator (e.g., Figure 12). An example is pictured in Figure 13. We postulate that the result is a non-monotonic convex function; i.e., one with a unique minimum, and thus suitable for internal exploration by any of a number of local search procedures which operate on smooth functions. However, this remains to be proven.

4.3.1 Conjugate Directions

The internal search employed is the *conjugate direction* method of Powell (1964; see Brent, 1973) as programmed in Press et al. (1989). It performs a series of $O(d^2)$ line minimizations where the last d directions are mutually orthogonal. An example 2-dimensional search is depicted in Figure 16. The initial probe is at "`x`" (the mean of the simplex vertices) and the search lines are numbered in order. Convergence can be caused to occur arbitrarily close to the minimum of the function; still, this is only the internal model search for one simplex and needn't be concluded at high accuracy.

The wild excursion in Figure 16, between lines 2 and 3, is the result of doubling the step from $\underline{x}_0$, where the search began, to $\underline{x}_2$, the end result of the 2^{nd} (d^{th}) line search. As its result was poor, that line of inquiry was abandoned in favor of the original basis functions (the unit vector directions). A more refined version of Powell's method (e.g., Brent, 1973) might instead perform a line search along a direction normal to $(\underline{x}_2 - \underline{x}_0)$, and converge more rapidly.

Alternatively, a method employing gradients, and even Hessians, could be employed, as the internal model function (16) is analytic. This issue will be explored in future work to reduce the overhead calculations of the algorithm.

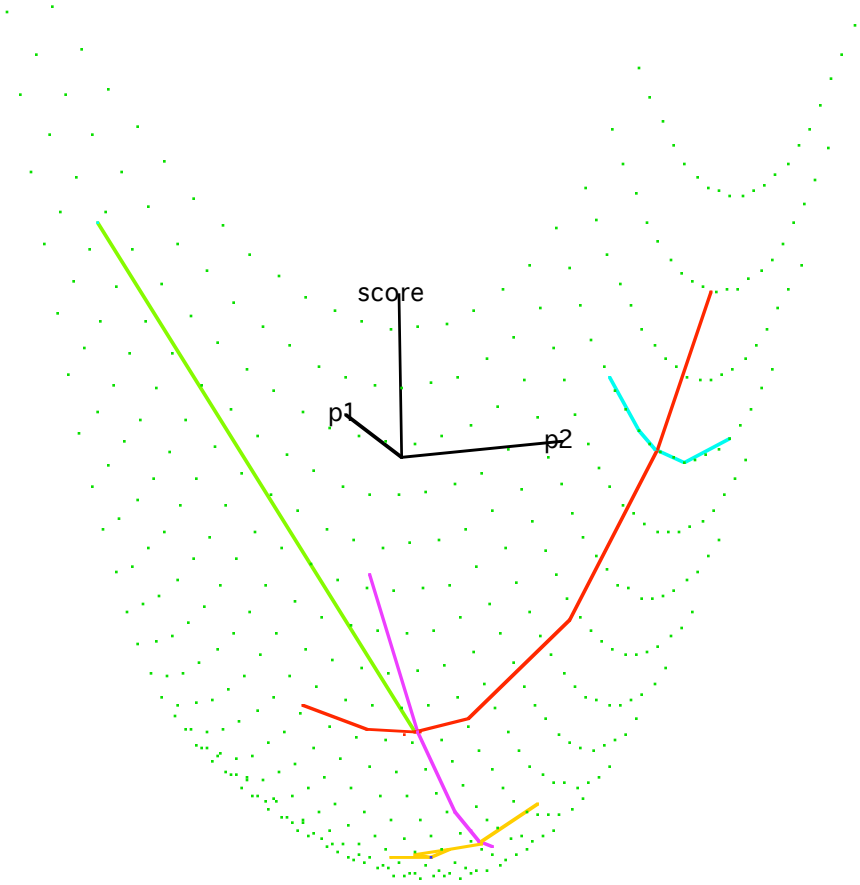


Figure 16: Example Internal Search Trajectory

4.3.2 Compositional Mapping

In the 2-dimensional prototype of the algorithm, during the internal search, each point was checked for membership in the simplex before the distance estimate (16) was calculated. That is, a point $\underline{x}$ was evaluated on if, for every face (line segment) k (k = 1, d+1), it was on the same side as the excluded vertex, v_k. If not, a large value was returned, with a component proportional to the distance of the point outside the triangle, to push the search back to the interior.

However, the proportion of outside points increases somewhat as the search progresses and the simplices shrink. Though not a problem in the 2-dimensional tests, it was feared that in higher dimensions, where the relative content of a simplex would be quite small, the problem could worsen. For instance, a d-sphere, which has considerably more content than an

equilateral *d*-tetrahedron (simplex), decreases in relative content with respect to a *d*-cube (having the same radius) surprisingly rapidly. The ratio is

$$\frac{\pi^{d/2}}{d\,2^{d-1}\,\Gamma(d/2)} \tag{21}$$

(e.g., Scott, 1992), which begins (1, 0.785, 0.524, 0.308, 0.164, 0.081, 0.037) for dimensions 1 through 7. The rapid decrease with *d* is such that (in a simulation test of this ratio), we found that only about one in a million random points in a 16-cube are also in the inscribed 16-sphere. This aspect of the "curse of dimensionality" indicates the potential danger of many points being outside the simplex of interest (adding to the overhead burden of the internal search) if performing a search in the original space.[20] Accordingly, an alternate approach was sought wherein any point could be mapped to a valid point and thus avoid checks. Techniques from the analysis of *compositions* provided the answer.

When analyzing components of a fixed whole (such as the in the introductory problem of finding the percentages, p_j, of assets to allocate to the classes: stocks, bonds, and cash) conventional techniques (e.g., regression) cannot be employed accurately as the variables $\underline{p}$ have a constraint of summing to unity. Instead of R^3, the valid space is the 2-dimensional surface of the unit simplex. Converting to a 2-dimensional equilateral triangle with unit height allows a composition $\underline{p}$ to be represented by the point which is p_j away from face$_j$ ($j = 1, d+1$).

Aitchison (1986) showed that transforming the $d+1$ variables by means of log-ratios, e.g.,

$$z_1 = \log\left(\frac{p_1}{p_3}\right)$$
$$z_2 = \log\left(\frac{p_2}{p_3}\right) \tag{22}$$

---

[20]This serves to caution application of intuition, trained on low dimensions, to higher-dimensional spaces.

produces d independent, unbounded variables, $\underline{z}$, amenable to classical analysis techniques. This mechanism, we realized, could also be used to map the bounded simplex region of an internal search into R^d, allowing conventional searches to be used there more confidently.

Points are called for as part of the search in the space **Z**; the location $\underline{z}$ is first mapped to a unit simplex through the inverse transformations ($j = 1, d+1$):

$$p_j = \frac{\exp(z_j)}{\sum_{i=1}^{d+1} \exp(z_i)} \tag{23}$$

where z_{d+1} is defined to be 0. These "corner proportions" $\underline{p}$ are then stretched over the particular triangle to get an actual location, $\underline{x}$

$$x_j = \sum_{i=1}^{d+1} p_i v_{ij} \tag{24}$$

where v_{ij} is the location of the i^{th} vertex along dimension j. The resulting score, $D(\underline{x})$ is then attributed to $\underline{z}$ for the purposes of the search.

The search paths of Figure 16 are shown in Figure 17 for **Z** and Figure 18 for **X**. In the latter, the nonlinearity of (23) is only faintly evident as the distance spanned by the local search was small relative to the size of the simplex. In general, a line in **Z** (e.g., one representing a consistent ratio between two components) is an arc in **X**.

The representation has other conveniences. The mean of the vertices in **X** (a convenient internal starting location) is represented as $\underline{0}$ in **Z**. The unreachable vertices themselves (mapped to in **Z**) are to be avoided anyway in the noiseless case, as they have response value ($\sigma^2(\underline{x})$ at the vertices is 0). Though points on the surface of the simplex are unreachable (there, at least one component, p_j, is 0, leading to $z_j =$) probes can be driven arbitrarily close by the **Z**-search. Only on the outer boundary of the entire search space, **A**,

will it matter that a probe is not exactly on a border, but this (and more) is compensated for by employing "attractive boundaries", as discussed in 4.2.

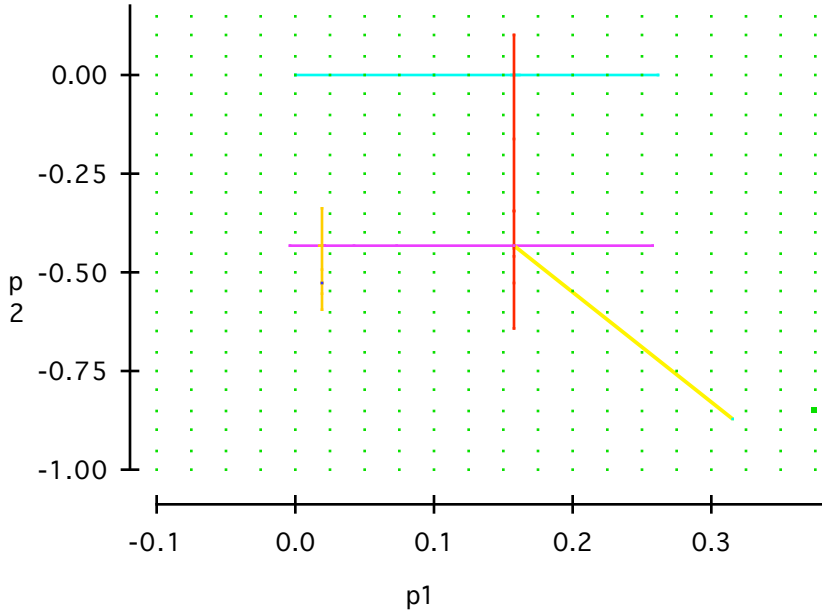


Figure 17: **Z**-Space Path of Search of Figure 16

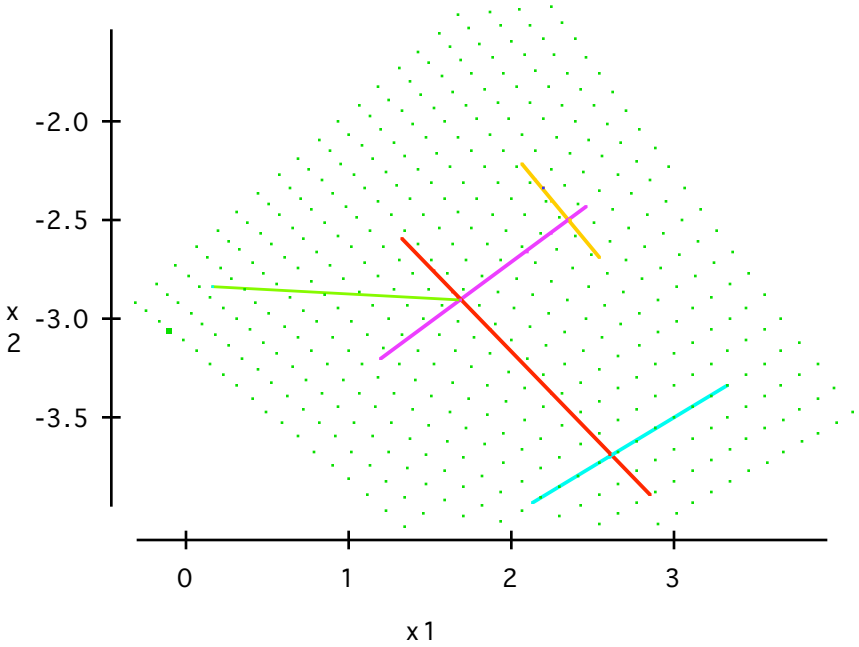


Figure 18: **X**-Space Path of Search of Figure 16

4.4 Goal Scheduling

Under relatively weak assumptions about the response surface, Chapter 3 showed it is possible to estimate the probe location having the highest probability of exceeding the current response goal. However, that choice is only *one-step optimal*; in general, the user has available a (short) sequence of probes, and would prefer to optimize the expected information return of the set.

Human-directed searches, it has been observed (Mockus, 1989), typically involve three stages: Exploration, Focusing, and Confirmation. First, a broad outline of the response is sought; in the vocabulary of this report, probes are selected primarily to eliminate variance in the response model. Secondly, promising regions are probed more finely. After these local searches, analysts tend to poke around a bit in some of the more "remote" regions, perhaps to gamble their final resources on finding some hidden treat or, at the least, to build confidence in the results already obtained.

Salient features of this process -- which is usually superior to algorithmic searches in the low dimensions accessible to visualization (op cit.) -- may be duplicated by scheduling our key parameter: the goal, Y_g. A high initial value emphasizes variance reduction; lower later values instead reward expectation. That is, far goals lead to candidate probes in unexplored areas (near the middle of large triangles), and near goals lead to probing regions with already minimal results -- a type of (gradient-free) local search.

Without information from the user about the function under study (which, in practice, may not be available or easily parameterized), the algorithm must discern what a reasonable goal is from the pattern of responses as they are obtained. The algorithm, in effect, gradually

adjusts its goals to accord with reality.[21] The user only provides a number, P_{max}, informing the program of its probe budget.

The mechanism selected for (virtually parameter-free) adjustment of the goal is to employ a monotonically decreasing multiplier, $\gamma(i)$, which exponentially relaxes from a high of 10.0 to a low of 0.1 (two orders of magnitude) over the course of the time allotted for internal probes (P_{max} - 2^d initializing probes) . This gain factor adjusts a measure of the latest *span* of observed responses, $Y_{max}(k,i)$ - $Y_{min}(i)$, where the first term is the k^{th} largest response after probe i. (This allows the goal to adjust more rapidly if extreme values exist, but are rare.) Every d+1 probes, the goal is set to

$$Y_g(i) = Y_{min}(i) - \gamma(i)*span\ (i) \tag{25}$$

The goal is not adjusted before every probe, as any change requires re-calculation of the candidate location and distance in every simplex, which is an $O(d^2)$ process for each simplex. A sample goal schedule for a test run is shown in Figure 19.

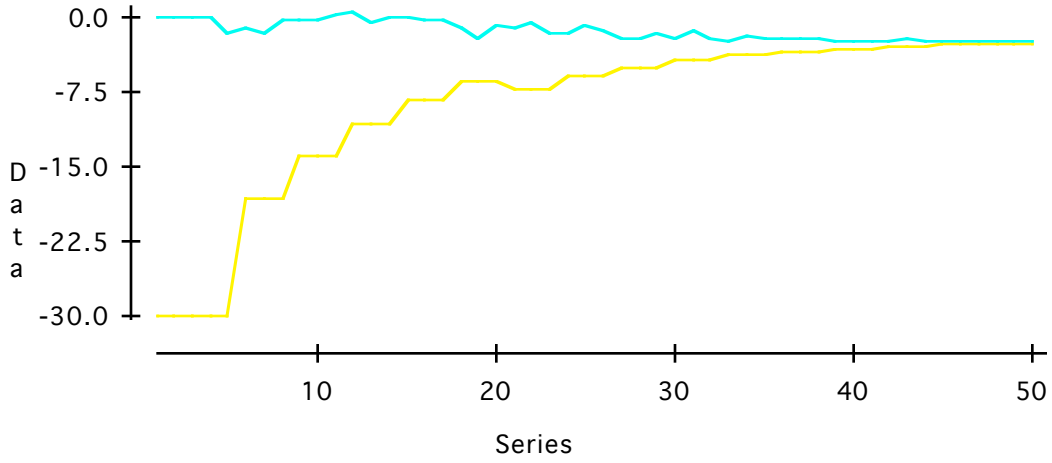


Figure 19: An Example Goal Schedule:
Y is the upper line, Y_g the lower

[21] Reminiscent of the bumper sticker, "If all else fails, lower your standards"! Actually, successfully negotiating such an adjustment process in the course of one's life has been found to be a strong contributor to overall satisfaction and happiness (Gallagher, 1993).

The exact internal parameter values in (25) are not essential; a strength of the process is its ability to adjust to the particulars of the function being explored. Several other tactical scheduling approaches have been investigated, including goals which are: constant, exponentially decreasing, led by an amount varied randomly within the bounds of the observed responses, based on a finite-state mechanism activated by the change in response values, and entered interactively by the user. A wide latitude of approaches should essentially *work*, but (25) seems the most consistent and convenient for a user of the method.

5. Experimental Results

By assumption, probes are resource-consuming to obtain for GROPE problems, so the chief criterion of merit in algorithm comparison on known problems will be the number of probes required to come very near the global optimum. (Though not realistic in practice, as the global optimum is not generally known, most reported results have employed this metric for the standard problems.[22]) The best known point cannot be "close enough" unless it is well inside the *basin of attraction* of the global minimum; that is, when a fine-stepped local search begun there will find $\underline{x}^*$. (Again, this information is unavailable in practice.)

Although the test problems employed here to facilitate comparison with other results are all quite simple to evaluate, a function evaluation in a practical application may involve running a complex simulation or performing a physical experiment. The new algorithm virtually disregards overhead costs then, to continually identify the current best set of potential probes to consider, given all that been found out about the response surface. However, at some point, those costs (time and/or memory requirements) will become onerous, so a preliminary experimental quantification of the order of complexity is provided.

5.1 Two-Dimensional Problem Suite

Most problems from the literature are two-dimensional, perhaps because they can be visualized and thus allow the progress of algorithms to be interpreted directly. A suite of eight common functions, recently collected by Cox and John (1992), are used as the primary test bed for the algorithm. The functions vary in their range of response values,

---

[22]The preferred criterion of converging with a given level of confidence in the results is unfortunately less frequently employed in the literature.

degree of smoothness, number of local and global minima, size of basins of attraction, amount of global and local *structure*, and level of complexity.[23]

For each function, results from the literature are listed (in chronological order)*when the algorithms are stopped near the global minima* (or for unknown reasons). Those techniques which seemed to use a more realistic convergence criterion are marked. (Note that when such runs are successful, they will take longer, so the number of probes required to reach $\underline{x}^*$ are fewer than reported; however, they do not always terminate at $\underline{x}^*$!) If more than one count is presented, they represent ranges or runs with different search parameters.

The current algorithm (especially while under development) has several exploratory parameters, including six different ways to schedule the search goal (4.4). For the runs reported here, the most robust version was employed. That is, the user was required only to specify P_{max}, the maximum number of probes to employ. This is a strong positive property of the algorithm as, in practice, this resource-related parameter is usually easier for a user to provide than a setting specific to either the function or the algorithm. (For all eight functions, P_{max} was set to 30, though other values were examined and found to agree closely.) Also, since $\underline{x}^*$ for a majority of the functions is at the mean of the space, the option to probe the interior there first (4.1.1) was turned off. (Otherwise, those searches would halt unfairly early.)

---

[23]A function's complexity may perhaps be reasonably estimated by its "minimum description length" using standard mathematical notation; i.e., by its *Kolmogorov complexity*.

5.1.1 Hosaki

The bimodal "Hosaki" function (Bekey and Ung, 1974):

$$(1 - 8x_1 + 7x_1^2 - \frac{7x_1^3}{3} + \frac{x_1^4}{4})\, x_2^2 \exp(-x_2) \qquad (26)$$

is shown in Figure 20. The global minimum for $x_1 \in [0,5]$, $x_2 \in [0,6]$ is -2.345 at location $\underline{x}^* = (4,2)$. Table 1 compares results.

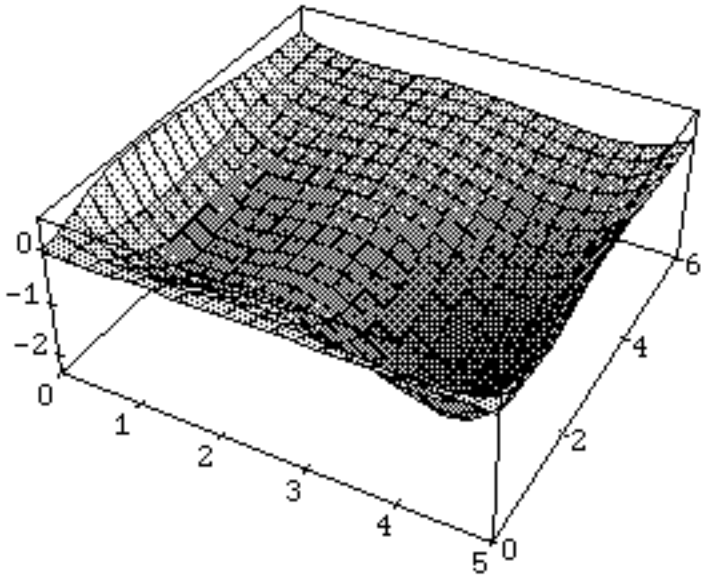


Figure 20: Hosaki Function

Table 1: Hosaki Results

Method	Reference	Probes
Modified Random Creep	Bekey & Ung, 1974	451
Adaptive Random Search	Pronzato et al., 1984	830
Sequential Design for Optimization	Cox & John, 1992	36, 55
GROPE	-	**27**

This function was the first explored by a prototype of the algorithm (Elder, 1992). The first trial employed the actual minimum, F(x*), as the (constant) goal, Y_g, but failed to converge in a reasonable time.[24] Graphical examination of the early progress of the search suggested it was too cautious in approaching the minimum; i.e., that local probing was overly preferred to exploration of new areas or, in other words, that the role of variance was too minor relative to that of expectation. Accordingly, a more remote goal, Y_g = -3.0, was set, leading to a much improved result: only 12 probes (with a final value of -2.344). The penultimate triangulation of this run is pictured in Figure 21, where each "x" represents the candidate probe location for its Delaunay triangle; the "." denote discarded candidate locations (due to dissolution of the surrounding triangle); and "-", the position of the global minimum. (The number of triangles and thus, candidate locations, increased by two after each probe -- a property of the 2-d Delaunay triangulation.[25]) The piecewise planar expectation surface of the resulting internal approximation is pictured in Figure 22 (shown inverted and with hidden lines removed, for clarity).

---

[24] It converged extremely slowly, getting 0.29 away from x* (and 0.045 over F(x*)) after the 24th probe, but taking another couple of hundred to (inexorably) get within a distance of 0.1.

[25] An exception not mentioned in the literature we examined, was discovered however: if the probe is exactly on an outer bound of the space, **A**, fewer new triangles are created. For example, in two dimensions, the total increments by 1 instead of 2.

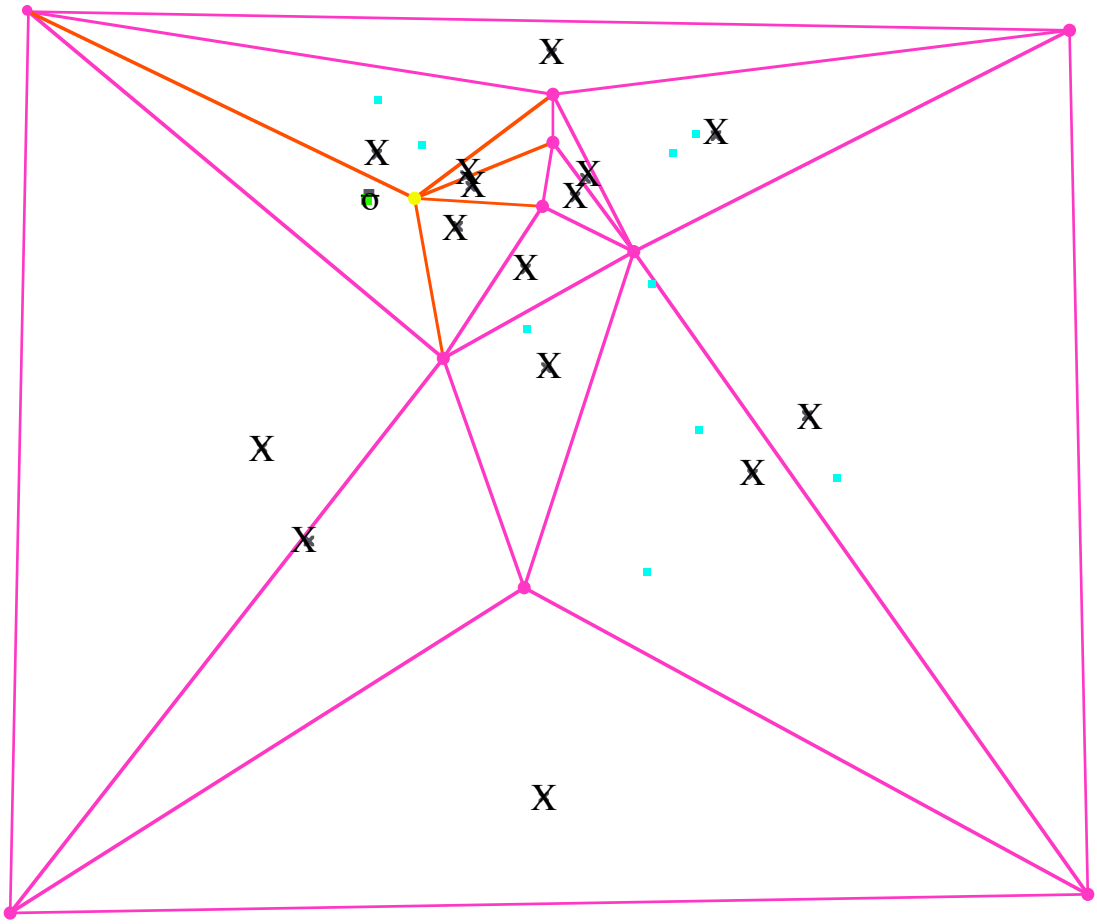


Figure 21: Triangulation after 11th Probe

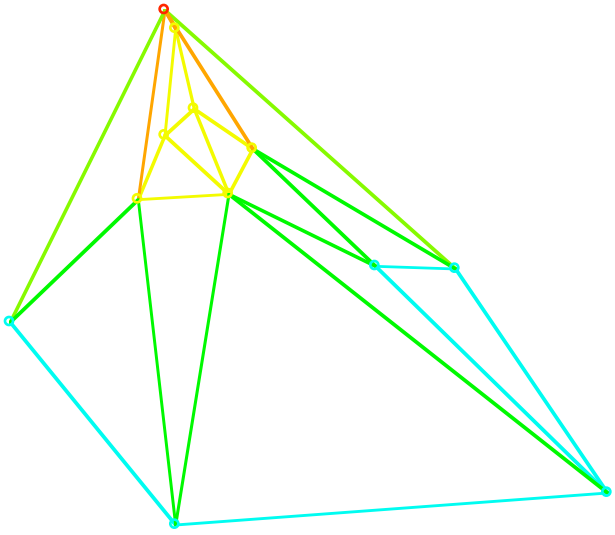


Figure 22: (Inverted) Expectation Surface due to the Triangulation of Figure 21

5.1.2 Basin 1

Three similar functions were employed by Bohachevsky et al. (1986) to test a *Simulated Annealing* algorithm. The first, shown in Figure 23, is:

$$2x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \qquad (27)$$

For all three, the global minimum for $x_1 \in [-1, 1]$, $x_2 \in [-1, 1]$ is $\underline{0}$ at location $\underline{x}^* = (0, 0)$.

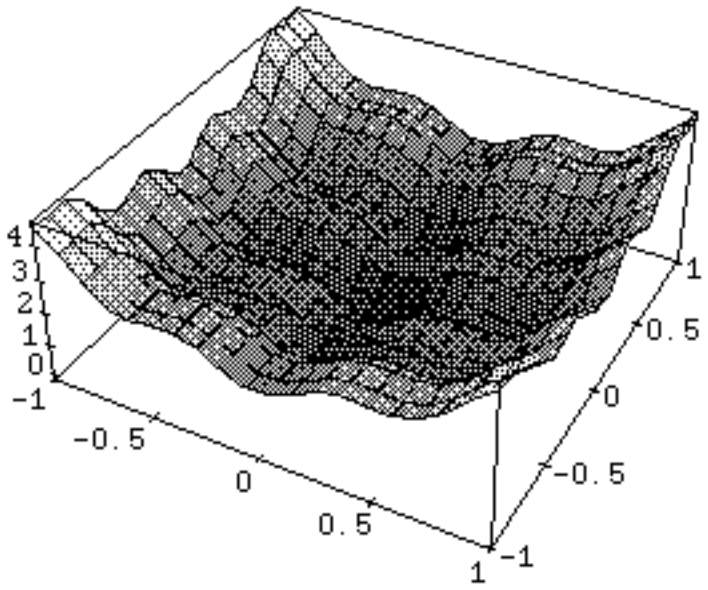


Figure 23: Basin.1 Function

Table 2: Basin.1 Results

Method	Reference	Probes
Simulated Annealing	Bohachevsky et al., 1986	150
Stochastic Probing	Laud, Berliner & Goel, 1989	66[c+]
Sequential Design for Optimization	Cox & John, 1992	**17**
GROPE	-	26

[c](convergence criterion)

+(median of a range of runs)

5.1.3 Basin 2

$$2x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3 \quad (28)$$

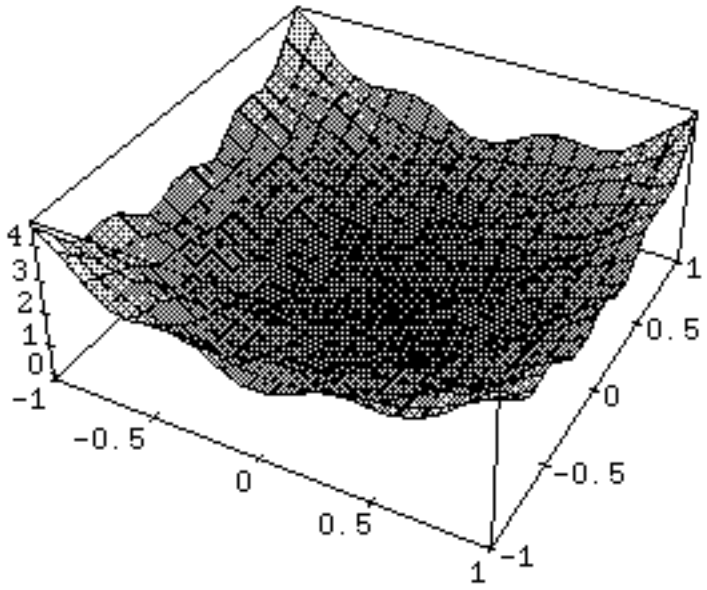


Figure 24: Basin.2 Function

Table 3: Basin.2 Results

Method	Reference	Probes
Simulated Annealing	Bohachevsky et al., 1986	400
Stochastic Probing	Laud et al., 1989	66[c+]
Sequential Design for Optimization	Cox and John, 1992	**17**
GROPE	-	26

[c](convergence criterion)

+(median of a range of runs)

5.1.4 Basin 3

$$2x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3 \quad (29)$$

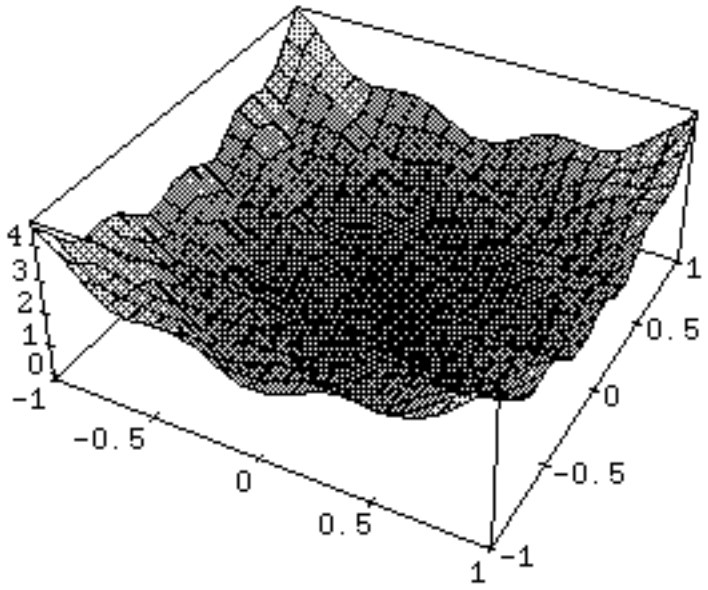


Figure 25: Basin.3 Function

Table 4: Basin.3 Results

Method	Reference	Probes
Simulated Annealing	Bohachevsky et al., 1986	400
Stochastic Probing	Laud, Berliner & Goel, 1989	100[c+]
Sequential Design for Optimization	Cox & John, 1992	**17**
GROPE	-	26

[c](convergence criterion)

+(median of a range of runs)

5.1.5 Sines

The biggest challenge for GROPE had pronounced local structure and a faint global trend:

$$1 + \sin^2(x_1) + \sin^2(x_2) - 0.1\exp(-x_1^2 - x_2^2) \qquad (30)$$

The global minimum for $x_1 \in [-10, 10]$, $x_2 \in [-10, 10]$ is $\underline{0.9}$ at location $\underline{x}^* = (0, 0)$. The other 40 or so local minima bottom out at 1.0. Our algorithm relatively quickly discovered several of the minima but had to be allowed many more probes than usual to find the global one. After spending much of its time exploring the details of many of the local minima, the probes got too dense there (removing most of the variance), and the method essentially turned to a "confirmation" phase, which eventually led to correct convergence.

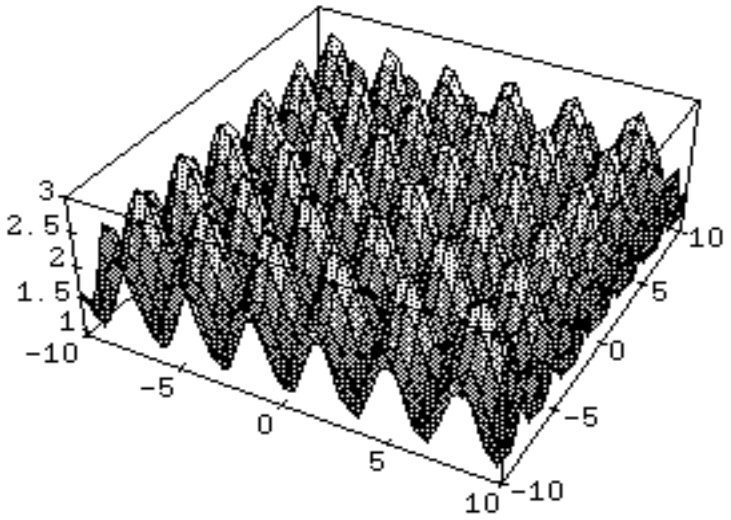


Figure 26: Sines Function

Table 5: Sines Results

Method	Reference	Probes
Controlled Random Search	Price, 1977	700
Sequential Design for Optimization	Cox & John, 1992	**17**
GROPE	-	352

5.1.6 3-Hump Camel-Back

$$2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} - x_1x_2 + x_2^2 \qquad (31)$$

The global minimum for $x_1 \in [-3, 3]$, $x_2 \in [-1.5, 1.5]$ is $\underline{0}$ at location $\underline{x}^* = (0, 0)$.

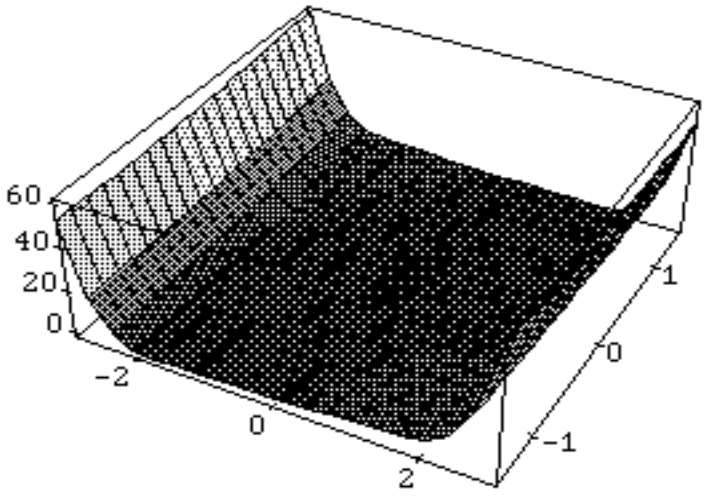


Figure 27a: 3-Hump Camel-Back Function

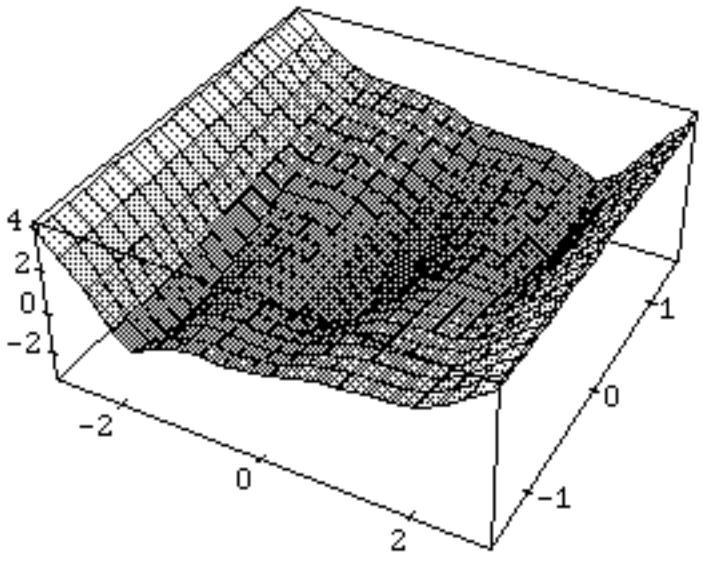


Figure 27b: Log(3-Hump Camel-Back) Function

Table 6: 3-Hump Camel-Back Results

Method	Reference	Probes
Trajectory	Hardy, 1975	1136
Adaptive Random Search	Pronzato et al., 1984	838
Sequential Design for Optimization	Cox & John, 1992	**17**
GROPE	-	24

5.1.7 Goldstein and Price

Very positive results were obtained on the (somewhat notorious) function

$$[1 + (x_1 + x_2 + 1)^2*(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$$
$$*[30 + (2x_1 - 3x_2)^2*(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)] \quad (32)$$

The global minimum for $x_1 \in [-2, 2]$, $x_2 \in [-2, 2]$ is $\underline{3}$ at location $\underline{x}^* = (0, -1)$, but the peak response value (a corner) is five orders of magnitude larger than those in the large neighborhood of $\underline{x}^*$. The function is shown in Figure 28a, and some detail in Figure 28b, but its structure is most apparent on the log scale of Figure 28c.

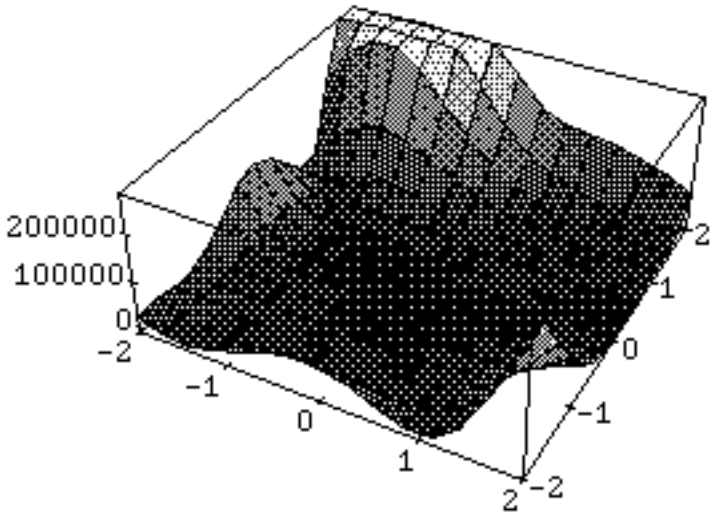


Figure 28a: Goldstein/Price Function

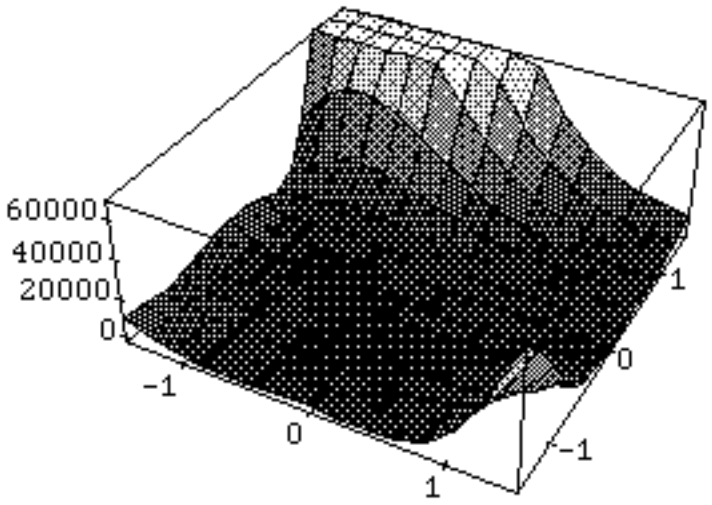


Figure 28b: Detail of Goldstein/Price Function

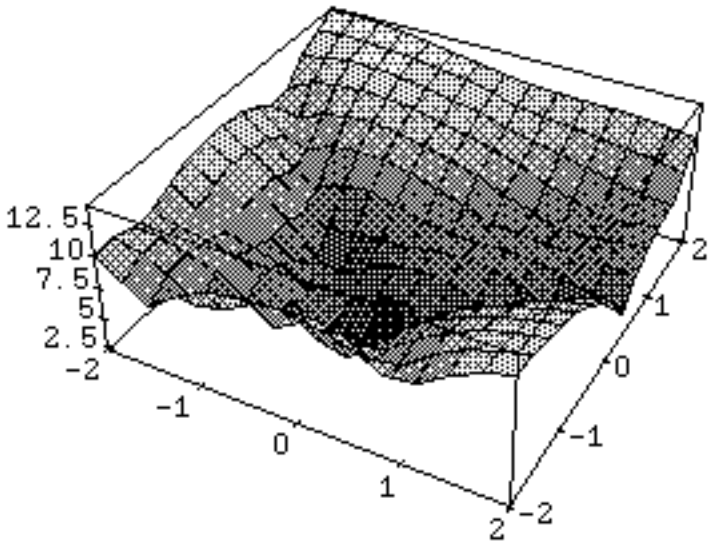


Figure 28c: Log(Goldstein/Price) Function

Table 7: Goldstein/Price Results

Method	Reference	Probes
Trajectory	Hardy, 1975	4242
Controlled Random Search	Price, 1978	2500
Space-Covering	Fagiouli, Pianca & Zecchin, 1978	1600
Density Clustering	Torn, 1978	2499
"	Gomulka, 1978b	1495
Adaptive Random Search	Pronzato et al., 1984	657
Self-Regulating Sim. Annealing	Vanderbilt & Louie, 1984	1186+
Multi-Start	Rinooy-Kan & Timmer, 1984	4400
Single Linkage Clustering	"	398
Multi-Level Single Linkage	"	294
Dynamic Multi-Start	Snyman & Fatti, 1987	474
Generalized Simulated Annealing	Brooks & Verdini, 1988	292
Stochastic Optimization	Mockus, 1989	362
"Hide & Seek" Sim. Annealing	Belisle, Romeijn & Smith, 1990	4728+
Multi-univariate Kushner	Perttunen & Stuckman, 1990	113
Diffusion Equation Method	Kostrowicki & Piela, 1991	120
Simultaneous Lipschitzian	Jones et al., 1992	101, 191
Sequential Design for Optimization	Cox & John, 1992	, 21*
Nonparametric Simplex	Perttunen, 1993	82
GROPE	-	**21**

*(most *SDO* runs halted with numerical errors)

+(median of a range of runs)

5.1.8 Branin

The final test function is due to Branin (1972):

$$\left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \qquad (33)$$

The global minimum for $x_1 \in [-5, 10]$, $x_2 \in [0, 15]$ is 0.3979 at *three* locations, $\underline{x}^* = \{ (-\pi, 12.25), (\pi, 2.25), (3\pi, 2.25) \}$. Again, the interesting details are hidden in the original scale (Figure 29a), but more visible in a log scale (Figure 29b).

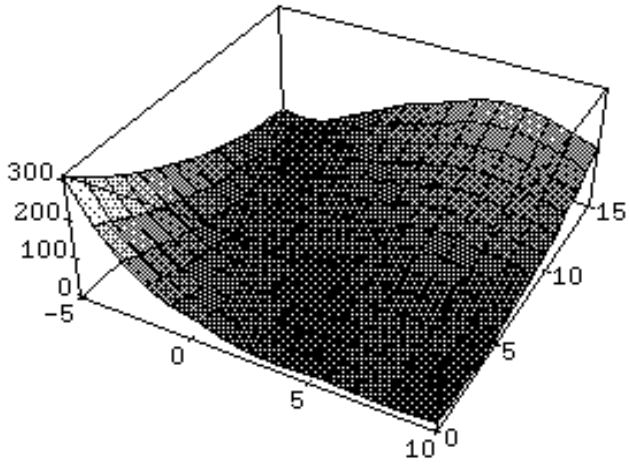


Figure 29a: Branin Function

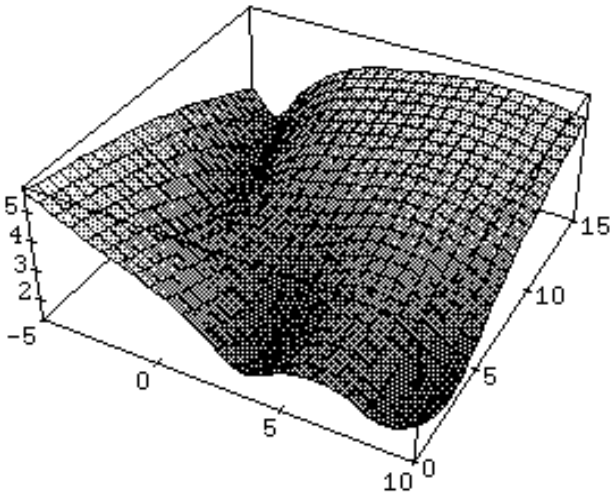


Figure 29b: Log(Branin) Function

Table 8: Branin Results

Method	Reference	Probes
Trajectory	Hardy, 1975	6005
Controlled Random Search	Price, 1978	1700+
Space-Covering	Fagiouli, Pianca & Zecchin, 1978	1600
Density Clustering	Torn, 1978	1558
"	Gomulka, 1978b	1318
"	DeBiase & Frontini, 1978	597c*
Self-Regulating Sim. Annealing	Vanderbilt & Louie, 1984	557+
Multi-Start	Rinooy-Kan & Timmer, 1984	1600
Clustering: Single Linkage	"	235
Clustering: *Multi-Level*	"	219
Generalized Simulated Annealing	Brooks & Verdini, 1988	292
Stochastic Optimization	Mockus, Tiesis & Zilinskas, 1978	1021
"	Mockus, 1989	189
"Hide & Seek" Sim. Annealing	Belisle, Romeijn & Smith, 1990	1846+
Multi-univariate Kushner	Perttunen & Stuckman, 1990	109
Simultaneous Lipschitzian	Jones et al., 1992	63, 195
Sequential Design for Optimization	Cox & John, 1992	172, 223
Nonparametric Simplex	Perttunen, 1993	97
GROPE	-	**29**

c(convergence criterion)

(note: did not stop at $\underline{x}$)

+(median of a range of runs)

5.2 Characterizing Appropriate Problems

Table 9 summarizes the results of the eight test functions, which fall into six groups. (The *Basin* problems (5.1.2 - 5.1.4) are very similar in shape and thus, results.) The best results are made bold, and the second best underlined. Note that the new algorithm places either first or second on each problem type; also, only *SDO* (Cox and John, 1992) improves on its results in the latter case. Thus, only the "best of the rest" method is noted for each function.

Table 9: Result Summary: Minimax Probes to Global Minimum

Function	GROPE	SDO	Other Method	
Hosaki	**27**	<u>55</u>	451	Modified Random Creep
Basin 1-3	<u>26</u>	**17**	100	Stochastic Probing
Sines	<u>352</u>	**17**	700	Constr. Random Sampling
3-Hump Camel	<u>24</u>	**17**	838	Adaptive Random Search
Goldstein/Price	**21**		<u>82</u>	Nonparametric Simplex
Branin	**29**	223	<u>97</u>	Nonparametric Simplex

The new algorithm and SDO dominate the field between them, and have complementary strengths and weakness.[26] Both these techniques, as well as the *Nonparametric Simplex* method (Perttunen, 1993) assume random fields and attempt to model the response surface. SDO employs a *global* model, where every probe result contributes something to the estimation of every internal trial point. Thus, it can pick up on global trends -- such as are especially evident in the first three problem classes of Table 9. However, such a global model is significantly affected by *outliers* when the scale changes drastically, as in the last two classes.

---

[26] E.g., a strategy which alternated probes between the two (with no sharing of results) would win on each of the problems.

Conversely, our method employes a *piecewise* model, able to respond to local variations in the response with only minor global ramifications. This makes it capable of addressing problems with wide or sudden variations in scale. Such "cliffs" are isolated by the piecewise model under the linear dominance property. When instead global function characteristics dominate, as in the first set of problems, this local model is adaptive enough to obtain good results in most cases (a first-place and three seconds out of four), even though it does not take full advantage of the global information available. The worst case for the assumptions is demonstrated by the *Sine* function, where there is a high ratio of *local noise to global trend*. Still, the algorithm proved robust enough to eventually provide the correct answer on that test problem.

Graphical summaries of the "learning curves" (log(#probes) vs. Date) the research community can be said to have travelled over the years are presented in Figures 30a and 30b for the last two test problems. (The data are from Tables 7 and 8, respectively). This view of the data suggests GROPE (the "x") is about two years "ahead of schedule".[27]

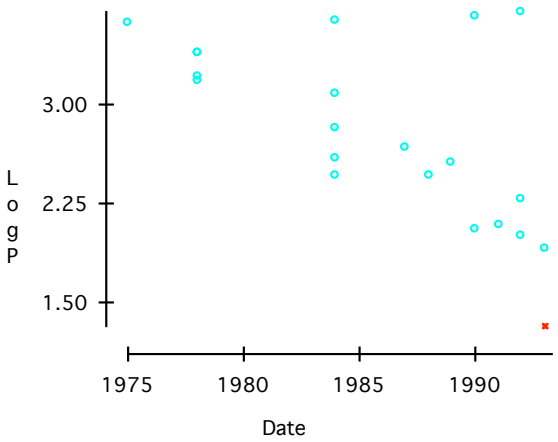


Figure 30a: Goldstein/Price Learning Curve

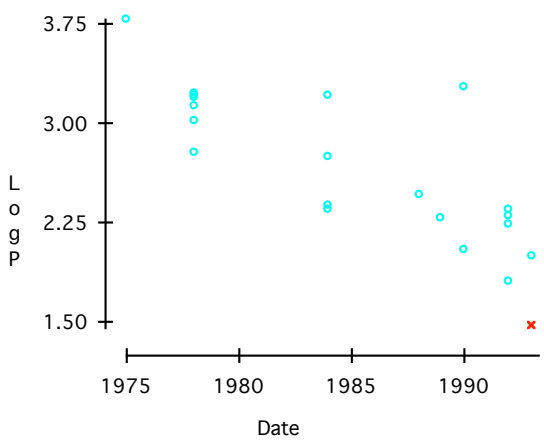


Figure 30b: Branin Learning Curve

[27] An observation from which the dissertation committee was asked not to draw too many conclusions!

5.3 Higher-Dimensional Timing Tests

Overhead calculations -- that is, time spent "thinking" and not probing, are quite extensive for our method, making it most appropriate for relatively low-dimensioned problems. To test the relative times required, functions of growing dimension were searched by the algorithm. Run times on an engineering workstation are presented in Table 10a for trials with constant goals, and Table 10b for scheduled goals. As the response functions were easy to calculate almost all of the time reported is for overhead.

Table 10a: Example Times with Constant Y_g (sec.)[28]

d	50	100	200	352
2	1.67	3.74	7.72	33.6
3	10.5	27.2	59.2	120.0
4	82.0	254.8	-	-
5	347.5	2043.	-	-

Table 10b: Example Times with Scheduled Y_g (sec.)

d	50	100	200	352
2	4.27	18.2	64.5	184.9
3	21.9	69.8	273.3	801.1
4	95.9	414.4	-	-
5	327.5	2032.	-	-

For constant goals with P probes, the computations are $O(Pe^d)$. When the goal is varied, the increased internal search calculations generally raise it to an $O(P^2e^d)$ process. Interestingly, for five dimensions, the scheduled-goal times are less than the constant-goal times for the relatively low number of probes allowed. This is likely because each internal search takes less time when the variance is high, as it is for the 2^d initial probes and several thereafter, when scheduling. When the variance is low, the final candidate probe location for each simplex is further from the simplex mean where the search is initialized.

---

[28] No scheduled run could be made to take longer than 352 probes on any of the test problems, so the constant goal run length was set to match.

6. Summary

The proposed model-based algorithm, which in large part generalizes Kushner's elegant univariate method, has the potential to become *the most efficient* optimizer for a significant subset of GROPE applications. For the low-dimensioned problems for which its overhead costs do not outweigh that of probing the system, the method should be more accurate with fewer probes than alternatives, and provide a measure of confidence in the outcome.

Using the typical test metric for known problems: "number of probes required to come very close to $\underline{x}^*$ (e.g., to within 0.01%)", the algorithm outperformed most competitors on a suite of standard functions. It was unbeaten on problems which may be characterized as having "local structure" determining the location of the global minima. It performed relatively poorly only in the case where the "global structure" of the response surface instead prevailed (and the local variations were a type of strong systematic noise). Still, it was either the most, or second-most, efficient algorithm for all the tests.

Notably, these results were obtained while treating the functions as "black boxes"; the user only provided a probe limit. When some information about the test function was instead employed (such as an estimate of the global function minimum, or an indication of how long to spend in an exploratory phase reducing variance) better results were obtained.

In all tests, one of the two recent algorithms based on a random walk model of the response surface (the other is SDO[29]) prevailed -- often requiring orders of magnitude fewer probes than conventional techniques. An intriguing project would be to explore how to merge, or smoothly parameterize, the distinctive properties of the approaches (which share many assumptions and goals) to produce a powerful hybrid method. Other suggestions for future research follow a brief summary of contributions.

---

[29]SDO is currently only encoded for two dimensions (Cox and John, 1992).

6.1 Contributions

The major contribution of this research is the design and implementation of a robust global optimization algorithm able to efficiently converge to multiple minima in a bounded space for a wide variety of GROPE problems. The method's chief practical strengths are the capability of its constantly-updated piecewise linear model to adapt to quite varied response surfaces, its need for very little information from the analyst about the system under study, and its ability to incorporate all known probes (even those from previous runs) in the estimation.

Additionally, a way to employ compositional mapping of *d*-simplex spaces to R^d to improve their exploration properties was described. This mechanism for dividing the search region into a set of simpler, unbounded search problems could find use in other optimization contexts, such as space-parallel searches.

6.2 Future Work

In addition to the need to prove convexity of the interior function (see Section 4.3) there are a number of interesting areas deserving exploration.

6.2.1 Convex Hull Initialization

The search boundaries of a larger class of problems could be more accurately represented by arbitrary linear constraints, rather than univariate limits. The boundary probes would be at the vertices of the convex hull, rather than *d*-rectangle, of the space, **A**, potentially requiring fewer probes (though more if the shape is complex).

The triangulation could be initialized by a simplex of "virtual" probes at locations extreme enough for the simplex space **S**, to contain **A**. (The virtual probes needn't require evaluation of the response function but could take on arbitrary (high) values.) The virtual probes could at the least simplify the triangulation, as all subsequent true probes would be interior to **S**,

and the empty circumsphere rule could be used to update the DT.[30] If the vertices of **A** are probed, then no candidate probe locations, etc. need be calculated for triangles having a virtual probe as a vertex.

But virtual probes could instead be employed to avoid explicit delineation of search boundaries.[31] The virtual probes would "nail down" the variance at the frontiers of the (vaguely) intended search space but, if high in imputed response value, would tilt the expectation inward to encourage interior probes. Thereby, the first function evaluations (true probes) could occur at the locations of most interest (highest prior probability of containing a minimum), such as near the mean of the space, rather than the (usually uninteresting and often arbitrary) bounds of A. Beginning real searching right away (rather than starting with 2^d corner probes) is more in line with the probe-preserving organizing principle of the algorithm. Still, considerable further research is required to learn how to modulate the values of the surrounding virtual probes to interfere with the search process at an appropriate level.

6.2.2 Noise

The model employed by the algorithm to maintain an updated estimate of the response surface is designed primarily to *agree with the probe results at the known locations*. This mechanism makes use of all known probes and insures that new locations are chosen in regions of sufficient uncertainty -- in effect, more thoroughly implementing the simple avoidance heuristic distinguishing the method of *Tabu* search (e.g., Glover, 1990). The estimation surface consists of a covering of piecewise planar regions defined by a simplex of probe responses. It generalizes the Markovian (memoryless) property of the univariate random walk model of Kushner (1962) by a linear analogy: points not defining the simplex (i.e., those linearly dominated by one of its edges) have no effect on its interior.

---

[30] Similar to a method used by Edelsbrunner (1993).

[31] or on regions of **A** where bounds are vague or frangible.

In the case of sampling a response function with noise σ^2, the variance "canopy" is merely raised that constant amount. However, the estimation component is much complicated, for now there is no justification for strict dominance of close but exterior points. Instead, some influence, fading with distance, should be allowed other points, in a manner perhaps similar to the related treatment by Cox and John (1992).

6.2.3 Stopping Condition

The search is to be terminated if either the goal or the probe limit is reached. Alternately, the slope parameter c (12) can be estimated from the results, and used to compute the probability, according to the underlying model, that some remaining location could exceed the goal. In a similar manner, Stuckman (1988) employs $\hat{c}$, the maximum likelihood estimator (MLE) of c, and the closest candidate probe, to get the probability that the *next* probe will do the job, stopping when this value is very small, e.g. 10^{-6}. However, as each probe is independent under the random walk model, one may calculate the *joint* probability that one of the N current candidates could exceed the goal

$$\Pr[\exists\, i \in y(p_i) > y_g \mid \mathbf{p}, D^2(\mathbf{p})] = 1 - \prod_{i=1}^{N} \Phi[\sqrt{D^2(p_i)}] \tag{34}$$

where $\hat{c}$ is used in $D^2()$ (14). A more reliable analysis of joint probability would be *predictive* (Aitchison and Dunsmore, 1975), and employ the full distribution of c. Such an estimator weights each possible c value by its relative effect on the likelihood of the data, $L(\mathbf{x}|c)$. In place of $D^2(\mathbf{x})$ with $\hat{c}$, one would use

$$\int_{C} D^2(\mathbf{x},c)\,p(c|\mathbf{x})\,\partial c \text{ , where } p(c|\mathbf{x}) = \frac{L(\mathbf{x}|c)}{\int_{C} L(\mathbf{x}|c)\,\partial c} \tag{35}$$

This is superior to the *estimative* technique of (34), which acts as if all the mass of the likelihood were on the MLE mode, $\hat{c} = \text{argmax}(L(\mathbf{x}|c))$. Computation would be substantially facilitated by using *conjugate priors* for the parameter distributions (op. cit.).

6.2.4 Parallel Computation

If q processors are available (and if the application permits) $q - 1$ probe locations, rather than just one, may be removed from the head of the ordered list and evaluated in Step 5 of the algorithm (Section 3.3). The last processor could update the Delaunay triangulation, given the locations of the new probes, as the triangulation does not depend on the probe results, **y**. Theoretically, the problem addressed could be reformulated to: find the best *set* of probes such that one is likely to exceed the goal (after Mockus, 1989). However, a simple q-at-a-time method should provide near-linear speedup in the early stages of many problems, when multiple regions of the domain have enough uncertainty to merit exploration.

6.2.5 Scaling

Like almost all search algorithms, the one introduced here assumes that the dimensions of the search region are scaled reasonably relative to one another; i.e., that the space is *homogenous*. In practice however, algorithms are often thrown directly at a problem, without the pre-processing or analysis that might help the search technique perform as advertised (or work at all). Accordingly, it would be wise to incorporate, into the planned public-domain encoding of this algorithm, some ability to diagnose or correct for scale inequities. Although the set of probe locations is dynamic and under the control of the algorithm, some results from scaling static data bases (e.g., Kruskal and Wish, 1978) should prove useful in this regard.

REFERENCES

Aarts, E.H.L., J. Korst (1989). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, New York.

Adler, R.J. (1981). *The Geometry of Random Fields*. Wiley, New York.

Aitchison, J. (1986). *Statistical Analysis of Compositional Data*. Chapman and Hall, Ltd., London.

Aitchison, J., I.R. Dunsmore (1975). *Statistical Prediction Analysis*. Cambridge Press, London.

Aluffi-Pentini, F., V. Parisi, F. Zirilli (1985). Global Optimization and Stochastic Differential Equations, *J. Optimization Theory and Applications* **47**: 1-16.

Aird, T., J. Rice (1977). Systematic Search in High-Dimensional Sets, *SIAM J. on Numerical Analysis* **14**: 296-312.

Andersson, R.S., P. Bloomfield (1975). Properties of the Random Search in Global Optimization, *J. Optimization Theory and Applications* **16**: 383-398.

Barker, A., D.E. Brown, W. Martin (1988). *A Neural Network Implementation of a Correspondence Processing Algorithm*. Inst. for Parallel Computation T.R. 88-008, Oct. 9.

Barron, R.L. (1968). Adaptive Flight Control Systems, in *Principles and Practices of Bionics*, Proc. AGARD Conf. **44**, Technivision Services, Slough, England (1970): 119-167.

Barron, R.L., R.L. Cellucci, P.R. Jordan III, N.E. Beam, P. Hess, A.R. Barron (1990). Applications of Polynomial Neural Networks to FDIE and Reconfigurable Flight Control, *Proc. National Aerospace Electronics Conf.*, Dayton OH, May 23-25.

Becker, R.W., G.V. Lago (1970). A Global Optimization Algorithm, *Proc. 8th Allerton Conf. on Circuits and Systems Theory*, Monticello, Illinois: 3-12.

Bekey, G.A., M.T. Ung (1974). A Comparative Evaluation of Two Global Search Algorithms, *IEEE Trans. Systems, Man & Cybernetics* **4**: 112-116.

Belisle, C., H. Romeijn, R. Smith (1990). Hide-and-Seek: A Simulated Annealing Algorithm for Global Optimization, *Tech. Report*. 90-25, Dept. Industrial and Operations Engineering, Univ. Michigan.

Betro, B., R. Rotondi (1984). A Bayesian Algorithm for Global Optimization, *Annals of Operations Research* **1**: 111-128.

Bock, M.E. (1992). Nonparametric Regression Using Wavelets, *Proc. 24th Symposium on the Interface: Computing Science and Statistics*, College Station, Texas, March 18-21.

Boender, C., A. Rinooy-Kan, G. Timmer, L. Stougie (1982). A Stochastic Method for Global Optimization, *Mathematical Programming* **22**: 125-140.

Bohachevsky, I.O., M.E. Johnson, M.L. Stein (1986). Generalized Simulated Annealing for Function Optimization, *Technometrics* **28**: 209-217.

Bowyer, A. (1981). Computing Dirichlet Tessellations, *The Computer Journal* **24**, no. 2: 162-166.

Branin, F.H. (1972). Widely Convergent Methods for Finding Multiple Solutions of Simultaneous Nonlinear Equations, *IBM J. of Research and Development*: 504-522.

Breiman, L., J.H. Friedman, R.A. Olshen, C.J. Stone (1984). *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, California.

Breiman, L. (1991). Hinging Hyperplanes for Regression, Classification, and Function Approximation, *Tech. Report*. 324, Dept. Statistics, UC Berkeley, CA.

Brent, R.P. (1973). *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Englewood Cliffs, New Jersey.

Brooks, D.G., W.A. Verdini (1988). Computational Experience with Generalized Simulated Annealing, *Proc. 20th Symposium on the Interface: Computing Science and Statistics*, Alexandria, VA.

Broomhead, D.S., D. Lowe (1988). Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems* **2**: 321-355.

Brown, D.E., C.L. Pittard, A.R. Spillane (1990). *ASSET: A Simulation Test Bed for Evaluating Data Association Algorithms*. Inst. for Parallel Computation T.R. 90-002, Jan. 22.

Cellucci, R.L., P. Hess (1990). Techniques for Developing and Applying Polynomial Network Synthesis Software, *Proc. 22nd Symposium on the Interface: Computing Science and Statistics*, East Lansing, MI, April.

Chaudhuri, P. M-C. Huang, W-Y. Loh, R. Yao (1990). Regression Trees and Function Estimation, *Tech. Report* 871, Dept. Statistics, Univ. Wisconsin-Madison, Dec.

Chichinadze, V.K. (1969). The Ω-Transform for Solving Linear and Nonlinear Programming Problems, *Automatica* **5**: 347-355.

Chichinadze, V.K. (1991). Solution of Nonlinear Nonconvex Optimization Problems by the Ω-Transform Method, *Computers and Mathematics with Applications* **21**, no. 6-7: 7-15.

Cleveland, W.S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots, *Journal of the American Statistical Association* **74**: 829-836.

Cole, R., J.S. Salowe, W.L. Steiger, E. Szemerédi (1989). An Optimal Time Algorithm for Slope Selection, *SIAM J. on Computing* **18**: 792-810.

Cover, T.M., P.E. Hart (1967). Nearest Neighbor Pattern Classification. *IEEE Trans. on Information Theory* **13**: 21-27.

Cox, D.D., S. John (1992). A Statistical Method for Global Optimization, *Tech. Report*, Dept. Statistics, Univ. Illinois (Champaign), April.

Dennis, J.E., Jr., V. Torczon (1991). Direct Search Methods on Parallel Machines, *SIAM J. Optimization* **1**, no. 4: 448-474.

Derigs, U. (1985). Using Confidence Limits for the Global Optimum in Combinatorial Optimization, *Operations Research* **33**, no. 6: 1024-1049.

Dixon, L.C.W., G.P. Szego (1978). The Global Optimization Problem: an Introduction, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 1-15.

Dwyer, R.A. (1989). Higher-Dimensional Voronoi Diagrams in Linear Expected Time, *Proc. 5th Annual Symposium on Computational Geometry*, June: 326-333.

Edelsbrunner, H. (1993). Personal communication.

Edelsbrunner, H., N.R. Shah (1992). Incremental Topological Flipping Works for Regular Triangulations, *Tech. Report* R-92-1726, Dept. Computer Science, Univ. Illinois, Urbana.

Elder, J.F. IV (1985). *User's Manual: ASPN: Algorithm for Synthesis of Polynomial Networks* (4th Ed., 1989). Barron Associates Inc., Stanardsville, Virginia.

Elder, J.F. IV (1991). Global Regression of Probe Evaluations, *3rd International Workshop on Artificial Intelligence and Statistics* (extended abstract), Ft. Lauderdale, Florida, Jan. 2-5.

Elder, J.F. IV (1992). A Novel Efficient Global Search Algorithm for Multiple Dimensions, *Proc. 24th Symposium on the Interface: Computing Science and Statistics*, College Station, Texas, March 18-21.

Elder, J.F. IV, R.L. Barron (1988) Automated Design of Continuously-Adaptive Control: The "Super-Controller" Strategy for Reconfigurable Systems, *Proc. American Control Conf.*, Atlanta, Georgia, June 15-17.

Elder, J.F. IV, D.E. Brown (1993). Induction and Polynomial Networks, in *Advances in Control Networks and Large Scale Parallel Distributed Processing Models*, *Vol. 2*. Ablex, Norwood, New Jersey [in press].

Elder, J.F. IV, M.T. Finn (1991). Creating 'Optimally Complex' Models for Forecasting. *Financial Analysts Journal*, Jan./Feb.: 73-79.

Fagiuoli, E., P. Pianca, M. Zecchin (1978). A Mixed Stochastic-Deterministic Technique for Global Optimization, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 103-116.

Faigle, U., R. Schrader (1988). On the Convergence of Stationary Distributions in Simulated Annealing Algorithms, *Information Processing Letters* **27**: 189-194.

Fletcher, R., M.J.D. Powell (1963). A Rapidly Convergent Descent Method for Minimization, *Computer J.* **6**: 163-168.

Flood, M.M., A. Leon (1964). A Universal Adaptive Code for Optimization (GROPE). *Recent Advances in Optimization Techniques*, A. Lavi and T. P. Vogl (Eds.), Wiley, New York.

Floudas, C.A., P.M. Pardalos (1990). *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Springer-Verlag, New York.

Friedman, J.H. (1988). Fitting Functions to Noisy Scattered Data in High Dimensions, *Proc. Computing Science and Statistics: 20th Symposium on the Interface*, Reston, Virginia.

Gallagher, W. (1993). Mid-life Myths, *The Atlantic Monthly* **271**, no. 5, May: 51-68.

Ge, R.P., Y.F. Qin (1987). A Class of Filled Functions for Finding Global Minimizers of a Function of Several Variables, *J. Optimization Theory & Applications* **54**: 241-252.

Glover, F. (1990). Tabu Search: A Tutorial, *Interfaces* **20**: 74-94.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Massachusetts.

Gomulka, J. (1978). Two Implementations of Branin's Method: Numerical Experience, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 151-164.

Gomulka, J. (1978b). A User's Experience with Torn's Clustering Algorithm, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 63-70.

Groch, A., L.M. Vidigal, S.W. Director (1985). A New Global Optimization Method for Electronic Circuit Design, *IEEE Trans. Circuits and Systems* **32**, no. 2.

Hardy, J.W. (1975). An Implemented Extension of Branin's Method, in *Towards Global Optimization*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 117-142.

Hartley, H.O., R.C. Pfaffenberger (1969). Statistical Control for Non-linear Optimization. *Themis Report* **18**, Inst. Statistics, Texas A&M Univ.

Hastie, T.J., R.J. Tibshirani (1986). Generalized Additive Models, *Statistical Science* **1**: 297-318.

Hearne, L.B., E.J. Wegman (1992). Maximum Information Density Estimation using Random Tessellations, *Proc. 24th Symposium on the Interface: Computing Science and Statistics*, College Station, Texas, March 18-21.

Hess, P., D.W. Abbott (1988). *Multi-dimensional Search and Optimization with the OMNIsearch Algorithm*, Barron Assoc., Inc. Report for Century Computing, Inc., PO 5035 under USAF F33615-84-C-3609, August.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Horst, R., H. Tuy (1987). On the Convergence of Global Methods in Multiextremal Optimization, *J. Optimization Theory and Applications* **54**: 253-271.

Incerti, S., V. Parisi, F. Zirilli (1979). A New Method for Solving Nonlinear Simultaneous Equations, *SIAM J. Numerical Analysis* **16**: 779-789.

Ivakhnenko, A.G. (1968). The Group Method of Data Handling -- A Rival of the Method of Stochastic Approximation, *Soviet Automatic Control* **3**: 43-55.

Joe, B. (1989). Three-Dimensional Triangulations from Local Transformations, *SIAM J. Scientific and Statistical Computing* **7**: 514-539.

Joe, B. (1991). Construction of k-Dimensional Delaunay Triangulations Using Local Transformations, *Tech. Report*, Dept. Computing Science, Univ. Alberta, Edmonton.

Joe, B. (1993). Personal communication.

Jones, D.R., C.D. Perttunen, B.E. Stuckman (1992). Global Optimization: Beyond the Lipschitzian Model, *Proc. IEEE Systems, Man & Cybernetics Conf.*, Chicago, Illinois, Oct. 18-21.

Kostrowicki, J., L. Piela (1991). Diffusion Equation Method of Global Minimization: Performance on Standard Test Functions, *J. of Optimization Theory and Applications* **69**: 269-284.

Kruskal, J.B., M. Wish (1978). *Multidimensional Scaling*. Sage University Series on Quantitative Applications in the Social Sciences, no. 07-011, Sage Publications, Beverly Hills.

Kushner, H.J. (1962). Stochastic Model of an Unknown Function, *Journal of Mathematical Analysis and Applications* **5**: 150-167.

Kushner, H.J. (1964). A New Method of Locating the Maximum of an Arbitrary Multipeak Curve in the Presence of Noise, *J. of Basic Engineering,* March: 97-106.

Laud, P. L.M. Berliner, P.K. Goel (1989). A Stochastic Probing Algorithm for Global Optimization, *Proc. 24th Symposium on the Interface: Computing Science and Statistics*, Alexandria, VA.

Lauren, P.D., N. Nandhakumar (1992). Determination of the Relative Orientation of Projections of Asymmetric Objects, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June.

Lawson, C.L. (1972). Generation of a Triangular Grid with Application to Contour Plotting, *CIT Jet Propulsion Lab., Tech. Mem.* 299.

Levy, A.V., A. Montalvo (1985). The Tunneling Algorithm for the Global Minimization of Functions, *SIAM J. Scientific and Statistical Computing* **6**: 15-29.

Lorensen, W.E., H.E. Cline (1987). Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics* **21**: 163-169.

McMurty, G.J., K.S. Fu (1966). A Variable Structure Automation used as a Multimodal Searching Technique, *IEEE Trans. on Automatic Control* **11**: 379-387.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller (1953). Equation of State Calculation by Fast Computing Machines, *J. Chemical Physics* **21**: 1087-1092.

Miles, R.E. (1970). On the Homogeneous Planar Poisson Point Process, *Mathematical Biosciences* **6**: 85-127.

Mladineo, R.H. (1986). An Algorithm for Finding the Global Maximum of a Multimodal, Multivariate Function, *Mathematical Programming* **34**: 188-200.

Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwar Academic Publishers, Boston.

Mockus, J., V. Tiesis, A. Zilinskas (1978). The Application of Bayesian Methods for Seeking the Extremum, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 117-129.

Nelder, J.A., R. Mead (1965). A Simplex Method for Function Minimization, *Computer Journal* **7**: 308-313.

Niederreiter, H., P. Peart (1982). Optimization of Functions by Quasi-Random Search Methods, *Caribbean J. Mathematics* **1**: 27-44.

Pannullo, J.E. (1992). *Risk of Extreme Events: Reliability and the Value of Information*. Ph.D. dissertation, Dept. Systems Engineering, Univ. Virginia, Charlottesville.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode, *Annals of Mathematical Statistics* **33**: 1065-1076.

Perttunen, C.D. (1991). A Computational Geometric Approach to Feasible Region Division in Constrained Global Optimization, *Proc. IEEE Systems, Man, & Cybernetics Conf.*, Charlottesville, Virginia, Oct. 13-16.

Perttunen, C.D. (1993). Simplex-Based Global Optimization Algorithms Using Nonparametric Statistical Transformations, *IEEE Trans. on Systems, Man, & Cybernetics* [to appear].

Perttunen, C.D., B.E. Stuckman (1990). The Rank Transformation Applied to a Multiunivariate Method of Global Optimization, *IEEE Trans. Systems, Man & Cybernetics* **20**, no. 5: 1216-1220.

Pinkus, M. (1968). A Closed-Form Solution of Certain Programming Problems, *Operations Research* **16**: 690-694.

Press, W.H., B.P. Flannery, S.A. Teukolsky, W.T. Vetterling (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, New York.

Price, W.L. (1977). A Controlled Random Search Procedure for Global Optimization, *Computer J.* **20**: 367-370.

Price, W.L. (1978). A Controlled Random Search Procedure for Global Optimization, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 71-84.

Pronzato, L., E. Walter, A. Venot, J.F. Lebruchec (1984). A General Purpose Global Optimizer: Implementation and Applications, *Mathematics and Computers in Simulation* **26**: 412-422.

Rajan, V.T. (1991). Optimality of the Delaunay Triangulation in R^d, *Proc. 7th ACM Symposium on Computational Geometry*: 357-363.

Ratschek, H. (1985). Inclusion Functions and Global Optimization, *Mathematical Programming* **33**: 300-317.

Rinooy-Kan, A., G.T. Timmer (1984). Stochastic Methods for Global Optimization, *American J. Mathematical and Management Sciences* **4**: 7-40.

Seidel, R. (1982). The Complexity of Voronoi Diagrams in Higher Dimensions, *Proc. 20th Annual Allerton Conf. on Communication, Control, and Computing*, Univ. Illinois at Urbana-Champaign, Oct.: 94-95.

Scott, D.W. (1985). Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions, *Annals of Statistics*.**13**, no. 3: 1024-1040.

Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York.

Schagen, I.P. (1986). Internal Modeling of Objective Functions for Global Optimization, *J. Optimization Theory and Applications* **51**: 345-353.

Sibson, R. (1978). Locally Equiangular Triangulations, *Computer J*. **21**, no. 3: 243-245.

Skeppstedt, A., L. Ljung, M. Millnert (1992). Construction of Composite Models from Observed Data, *International Journal of Control* **55**, no. 1: 141-152.

Snyman, J., L. Fatti (1987). A Multi-Start Global Minimization Algorithm with Dynamic Search Trajectories, *J. of Optimization Theory and Applications* **54**: 121-141.

Sobol (1979). On the Systematic Search in a Hypercube, *SIAM J. Numerical Analysis* **16**: 790-793.

Solis, F.J., R.J-B. Wets (1981). Minimization by Random Search Techniques, *Mathematics of Operations Research* **6**: 19-30.

Specht, D.F. (1967). Generation of Polynomial Discriminant Functions for Pattern Recognition, *IEEE Trans. on Electronic Computers* **16**: 308-319.

Spendley, W., G.R. Hext, F.R. Himsworth (1962). Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation, *Technometrics* **4**: 441-461.

Stuckman, B.E. (1988). A Global Search Method for Optimizing Nonlinear Systems, *IEEE Trans. Systems, Man & Cybernetics* **18**, no. 6: 965-977.

Stuckman, B.E., P. Scannell (1991). A Multidimensional Bayesian Global Search Method which Incorporates Knowledge of an Upper Bound, *Proc. IEEE Systems, Man & Cybernetics Conf.*, Charlottesville, Virginia, Oct. 13-16.

Timmer, G.T. (1984). *Global Optimization: a Stochastic Approach*. Ph.D. dissertation, Erasmus Univ., Rotterdam.

Tipper, J.C. (1991). FORTRAN Programs to Construct the Planar Voronoi Diagram, *Computers and Geosciences* **17** no. 5: 597-632.

Torczon, V. (1989). *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. Ph.D. dissertation, Dept. Mathematical Sciences, Rice Univ., May.

Torn, A.A. (1977). Cluster Analysis using Seed Points and Density-Determined Hyperspheres as an Aid to Global Optimization, *IEEE Trans. Systems, Man & Cybernetics* **7**: 610-616.

Torn, A.A. (1978). A Search-Clustering Approach to Global Optimization, in *Towards Global Optimization 2*, L.C.W. Dixon and G.P. Szego (Eds.), North-Holland, Amsterdam: 49-62.

Torn, A.A., A. Zilinskas (1989). *Global Optimization*. Springer-Verlag, New York.

Vanderbilt, D., S. Louie (1984). Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables, *J. of Computational Physics* **56**: 259-271.

Walters, F.H., L.R. Parker, Jr., S.L. Morgan, S.N. Deming (1991). *Sequential Simplex Optimization*. CRC Press, Boca Raton.

Watson, D.F. (1981). Computing the *N*-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes, *The Computer Journal* **24**, no. 2: 167-172.

Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Dissertation, Harvard, August.

Wood, G.R. (1985). Multidimensional Bisection and Global Minimization, *Tech. Report*, Univ. Canterbury.

Zilinskas, A. (1976). The Method of One-Dimensional Multiextremal Minimization, *Engineering Cybernetics* **76**, no. 4: 71-74.

Zilinskas, A. (1982). Axiomatic Approach to Statistical Models and their use in Multimodal Optimization Theory, *Mathematical Programming* **22**: 104-116.

Appendix

Theorem 1

For a constant goal, Kushner probes introduce two equally appealing candidate probes.

Proof: Given known results at neighboring locations x_a and x_b ($x_a < x_b$) the intermediate probe location x_0 under Kushner's algorithm is $x_a + p(x_b - x_a)$, where p is $\frac{\sigma_a}{\sigma_a + \sigma_b}$, and $\delta^* = y^* - Y_g$, the distance from a response to the goal value, Y_g (Eqns. 11, 12, 15). Given the probe's response, y_0, the expectation and variance equations for the new left and right segments are

$$\mu_L(x) = y_a + p_L(x)(y_0 - y_a) \qquad \mu_R(x) = y_0 + p_R(x)(y_b - y_0)$$

$$\sigma^2_L(x) = p_L(1 - p_L)(x_0 - x_a) \qquad \sigma^2_R(x) = p_R(1 - p_R)(x_b - x_0)$$

where

$$p_L(x) = \frac{x - x_a}{x_0 - x_a} \qquad p_R(x) = \frac{x - x_0}{x_b - x_0}$$

By (15), the optimal proportions are functions only of y values:

$$p_L = \frac{\sigma_a}{\sigma_a + \sigma_0} \qquad p_R = \frac{\sigma_0}{\sigma_0 + \sigma_b}$$

Thus, the distances to the goal of the best points in the segments, x_L and x_R, are

$$\mu_L(x_L) - Y_g = \delta_a + \frac{\sigma_a(y_0 - y_a)}{\sigma_0 + \sigma_a} \qquad \mu_R(x_R) - Y_g = \delta_b + \frac{\sigma_0(y_b - y_0)}{\sigma_b + \sigma_0}$$

$$= 2\frac{\sigma_0\sigma_a}{\sigma_0 + \sigma_a} \qquad = 2\frac{\sigma_0\sigma_b}{\sigma_0 + \sigma_b}$$

and the variances there are

$$\sigma^2_L(x_L) = \frac{\sigma_a\sigma_0}{(\sigma_a + \sigma_0)^2}(x_0 - x_a) \qquad \sigma^2_R(x_R) = \frac{\sigma_b\sigma_0}{(\sigma_b + \sigma_0)^2}(x_b - x_0)$$

$$= \frac{\sigma_a^2\sigma_0^2(x_b - x_a)}{(\sigma_a + \sigma_0)^2 - (\sigma_a + \sigma_b)} \qquad = \frac{\sigma_b^2\sigma_0^2(x_b - x_a)}{(\sigma_b + \sigma_0)^2 - (\sigma_a + \sigma_b)}$$

causing the squared standardized distances to the goal, $\frac{(\mu(x) - Y_g)^2}{\sigma^2(x)}$ to be

$$D^2(x_L) = 4\,\frac{\sigma_0^2\sigma_a^2(\sigma_a + \sigma_0)^2(\sigma_a + \sigma_b)}{(\sigma_0 + \sigma_a)^2\sigma_a^2\sigma_0(x_b - x_a)} \qquad D^2(x_R) = 4\,\frac{\sigma_0^2\sigma_b^2(\sigma_b + \sigma_0)^2(\sigma_a + \sigma_b)}{(\sigma_0 + \sigma_b)^2\sigma_b^2\sigma_0(x_b - x_a)}$$

$$= 4\,\frac{\sigma_0(\sigma_a + \sigma_b)}{x_b - x_a}$$

So, the candidate probes at x_L and x_R are estimated to be equally likely to exceed Y_g.