Proceedings IEEE International Conference on Systems, Man, and Cybernetics, Chicago, Illinois, October 18-21, 1992

# Global $R^d$ Optimization when Probes are Expensive: the GROPE Algorithm

John F. Elder IV
Department of Systems Engineering
University of Virginia
Charlottesville, VA 22903-2442
jfe7w@Virginia.edu

*Abstract* **A global optimization algorithm is introduced which generalizes Kushner's univariate search [1]. It aims to minimize the number of probes (function evaluations) required for a given confidence in the results. All known probes contribute to a stochastic model of the underlying "score surface"; this model is interrogated for the location most likely to exceed the current result goal. The surface is assumed to be fractal, leading to a piecewise Gaussian model, where the local regions are defined by the Delaunay triangulation of the probes. The algorithm balances the competing aims of 1) sampling in the vicinity of known peaks, and 2) exploring new regions. Preliminary tests on a standard 2-d search problem are very encouraging.**

## I. INTRODUCTION

Global search can optimize parameters which nonlinearly affect the output of a model, such as with logistic regression, or the intermediate weights of an artificial neural network (ANN). Global search is also called for when the fitting criterion, or *score function*, is anything other than a few special accuracy metrics, such as minimum squared error (MSE, $L_2$) or least absolute deviations (LAD, $L_1$). For instance, the cost of errors can be asymmetric (e.g., a classification "false alarm" can be much less costly than a "false dismissal") or, a range of estimated values might correctly lead to the same action (e.g., "buy"), or the true score function might include objectives (low cost, high safety, etc.) other than accuracy . Arguably, linear $L_2$ models are employed so extensively not because their assumptions are so often appropriate, but due to their strong mathematical tractability and the mass of tools available as a result. The significant benefits of linear models should only reluctantly be abandoned, but *once set aside, the resulting freedom to design a score function to match the use of the model should be exploited.* This requires global search.

## II. MODEL-BASED SEARCH

Global optimization can be likened to depth-sounding (Fig. 1). One probes for the depth (evaluates the score function) at a location (set of parameter values) in pursuit of the deepest spot (global minimum). Model-based searches build up a picture of the ocean floor using 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 surface model 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.

The algorithm introduced here attempts to generalize an elegant 1-dimensional stochastic method [1][3] inspired by the work of Stuckman and colleagues [4]-[7]. Recent refinements are believed to lead to a more theoretically consistent (and hence more efficient) $R^d$ algorithm, requiring drastically fewer function evaluations than conventional searches. (It can be said to "think" more but "run" less.) The search 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, approximating the probability that the final answer can be improved upon. Lastly, the procedure can be parallelized, with nearly linear speed-up anticipated. This paper describes Kushner's 1-dimensional method, important later generalizations, and the new algorithm, known as Global $R^d$ Optimization when Probes are Expensive (GROPE).[1]

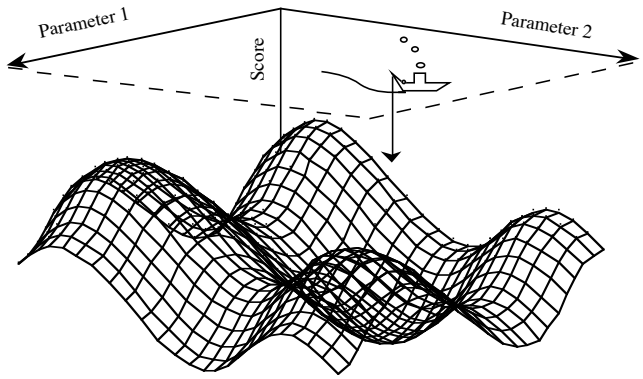



Fig. 1: Global Optimization as Depth-Sounding (after [2])

## III. 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 Fig. 2; the path is that of one moving forward at a constant rate, but staggering random amounts to each side. The stagger distribution is Gaussian, and for a time series with discrete steps, $y_{t+1}$ can be shown to be $N(y_t, \sigma^2)$. 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*.

[1] Earlier searches: Guided Random Optimizer of Performance Error [8] & Global Regression of Probe Evaluations [9]. Some acronyms never die!

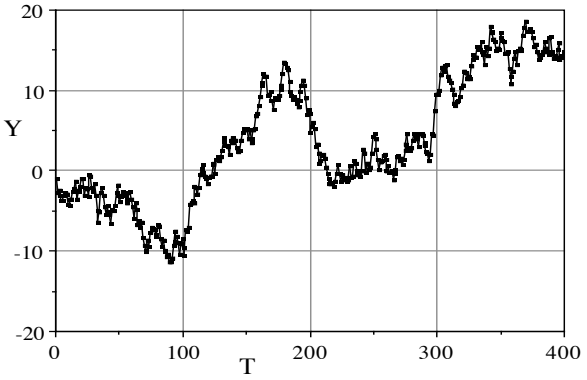



Fig. 2: An Example Random Walk: $Y = \sum X(t)$ , $X=N(0,1)$

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.

A further practical advantage of the representation is its tractability. 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

$$\mu(x \mid \mathbf{x},\mathbf{y}) = y_a + p(y_b - y_a) \quad (1)$$

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

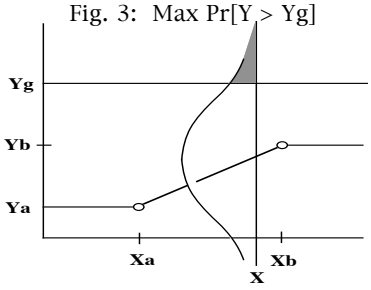
$$\sigma^2(x \mid \mathbf{x},\mathbf{y}) = cp(1 - p)(x_b - x_a) \quad (2)$$

for some slope factor, $c$ (the mean squared variation in $y$ as $x$ changes). (Note that $\sigma^2$ 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). When noise is present (i.e., probes at the same location can return different values), the representation is only slightly adjusted [3]: $\mu(x)$ does not go through the samples exactly, but *shrinks* toward neighboring samples, and $\sigma^2(x)$ is positive, not zero, at the probes.

Kushner [1] 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, predating the similar strategy of "temperature scheduling" which directs *simulated annealing* [21] A slightly different perspective [7] is to seek the point most likely to exceed a given *result goal*, $y_g$; i.e., to find the x maximizing

$$\Pr[y_x > y_g \mid \mathbf{x},\mathbf{y}] = 1 - \Phi\left[\frac{y_g - \mu(x|\mathbf{x},\mathbf{y})}{\sigma(x|\mathbf{x},\mathbf{y})}\right] \quad (3)$$

Fig. 3: Max Pr[Y > Yg]




This is depicted in Fig. 3 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, say, 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.

As $\Phi$ is monotonic, we may maximize (3) by *minimizing* its argument, (or square, to accommodate (2)). Substituting, this translates to finding the proportion $p$ which minimizes

$$A(p) = \frac{[y_g - (y_a + p(y_b - y_a))]^2}{cp(1-p)(x_b - x_a)} \quad (4)$$

Solving $\frac{\partial A}{\partial p} = 0$ reveals that the optimal location depends only on the relative distance of the end points to the goal $p^* = \frac{\Delta a}{\Delta a+\Delta b}$ where $\Delta a = y_g - y_a$, $\Delta b = y_g - y_b$. (Note that the slope parameter, $c$, has no influence on $p^*$, and may be dropped.) A(p*) is a distance monotonic with the segment's maximum conditional probability of containing a location exceeding the goal.

The 1-dimensional algorithm can be summarized:

1) Probe the two bounds of the search space.[2]
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, A($p^*$).
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.

Fig. 4 illustrates the algorithm for the data of Fig. 2.

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. (And it can be shown that their two distances are identical; e.g., Fig. 4b. In that example, ties were broken randomly.) When results are far from the goal (at the beginning of the search), the variance component of (4) dominates and locations relatively midway between known probes are preferred. When the best probes

[2] The initializing probes can be inside the bounds (for three line segments initially). However, probes may later be requested the edges of the legal space anyway, since $\sigma^2$ grows so rapidly beyond the outermost probe.

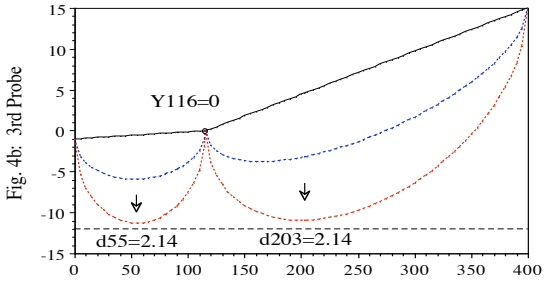



Fig. 4b: 3rd Probe

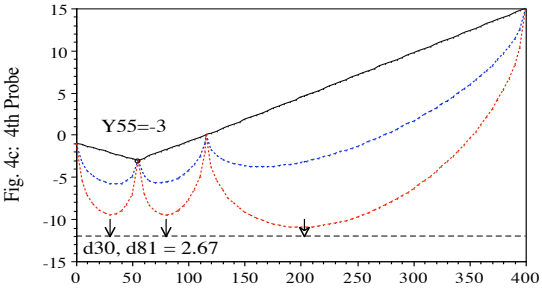



Fig. 4c: 4th Probe

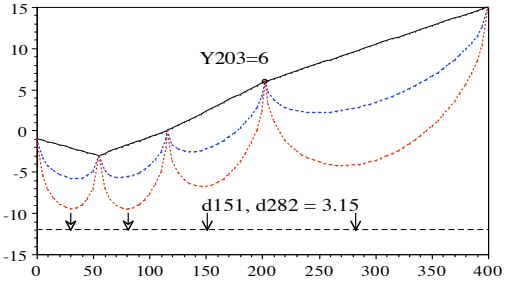



**Figure 4d: 5th Probe**

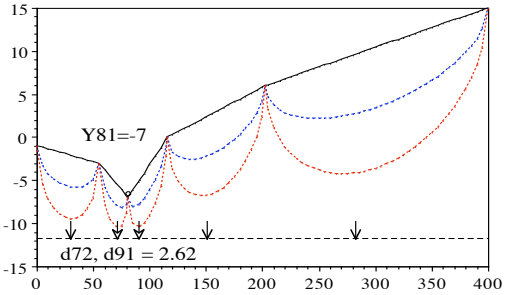



Fig. 4e: 6th Probe

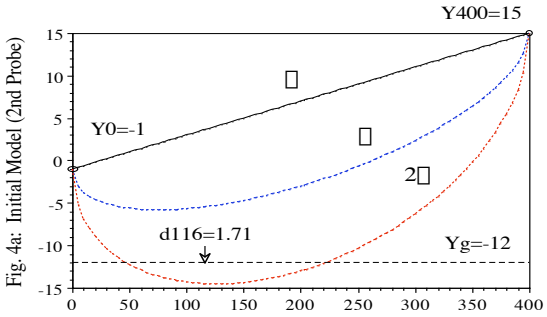



Fig. 4a: Initial Model (2nd Probe)

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.

The search is terminated if either the goal or the probe limit is reached. Alternately, the slope parameter $c$ 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. Similarly, [4] employs $\hat{c}$, the maximum likelihood estimator (MLE) of $c$, and the closest candidate probe, to get the probability that the *next* guess 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

$$\Pr[\exists\, i \in 1..N : y(p_i) > y_g \mid \mathbf{p}, \mathbf{A}(\mathbf{p})] = 1 - \prod_{i=1}^{N} \Phi[\sqrt{A(p_i)}] \qquad (6)$$

where $\hat{c}$ is used in A (4). A more reliable analysis of joint probability would be *predictive* [10], 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 $A(\mathbf{x})$ with $\hat{c}$, one would use

$$\int_C A(\mathbf{x},c)p(c|\mathbf{x})\partial c \text{ , where } p(c|\mathbf{x}) = \frac{L(\mathbf{x}|c)}{\int_C L(\mathbf{x}|c)\partial c} \qquad (7)$$

This is superior to the *estimative* technique of (6), which acts as if all the mass of the likelihood were on the MLE mode, $\hat{c} = \sup(L(\mathbf{x}|c))$. Computation is substantially facilitated by using *conjugate priors* for the parameter distributions [10].

## IV. EXTENSION TO MULTIPLE DIMENSIONS

The key difficulty in expanding Kushner's algorithm from $R^1$ to $R^d$ -- 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 [11]).

The *multi-univariate* method [4] 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 $K$ jumps an order of magnitude when the probe is the current best). However, such a procedure can ignore a probe intermediate to another pair and, more importantly, is silent about function values everywhere except on the connecting line segments.

To 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 $R^d$, 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., [12]), 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* [13] in which small angles in triangles are avoided. The unique set of connections with this property in the plane is the *Delaunay* triangulation [14]. (However, in three and higher dimensions, this triangulation does not necessarily maximize the minimum *solid angle* [15].)

The Delaunay triangulation is the dual of the *Voronoi* (or Dirichlet or Thiessen) tessellation, wherein regions are partitioned according to the nearest neighbor rule. 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 [16], but only recently proven in general [17]: the *circumscribing sphere* of each simplex is empty; i.e., the only triangulation in which the sphere intersecting the vertices of a given simplex contains no other point, is the Delaunay.

The optimization algorithm of [6] 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" [5]. 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). Incorporation of Delaunay triangulation improved the ranking method, causing the scatter plot of search points to better correspond to the contour diagram of each objective function tested [6]. However, use of the ranking, coverage, and weighting heuristics lead to a technique having little in common with Kushner's stochastic algorithm.

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 quadratic variance curve along each 1-dimensional simplex edge. The expectation is thus a piecewise planar surface – resembling facets of a gem (or, the recent *hinging hyperplane* modeling technique [18]). An example 2-dimensional Delaunay interpolation surface is illustrated in Fig. 5.

The relative variance "canopy" arches over the simplex as shown in Fig. 6, 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[3] (as with Kushner's 1-dimensional method); therefore, the edge constraints on variance are *necessary* for this generalization of the algorithm.

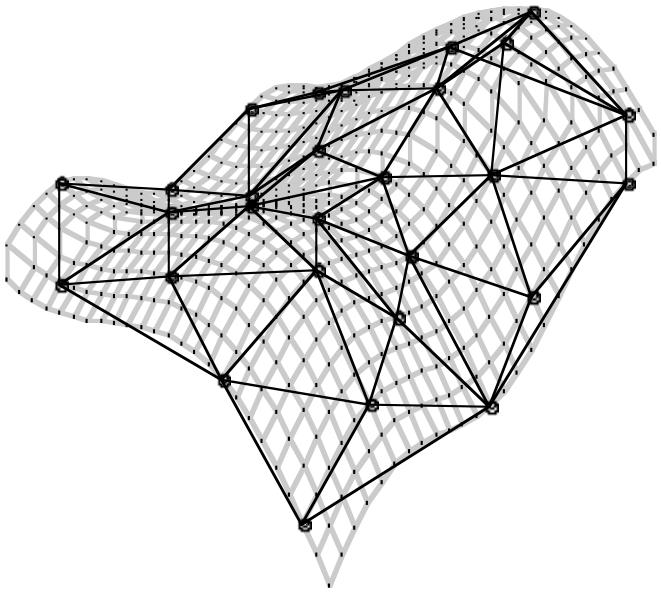


Fig. 5: Triangular Facets Interpolate Function Surface

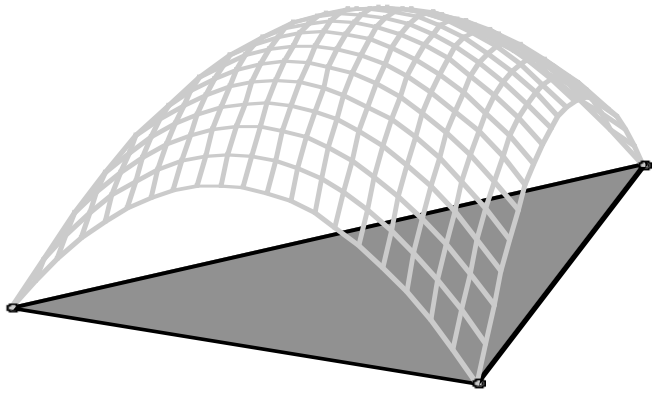


Fig. 6: Example Variance Canopy

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. (Still, in practice, thin triangles on the convex hull can lead to nearly collinear edges. Thus, robust regression techniques (e.g., singular value decomposition) which remove near-singularities are required.

[3] This 1-dimensional property can have the side-effect of ignoring the single 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.

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 (5). 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

$$A(\mathbf{x}) = \frac{(y_g - \mu(\mathbf{x}))^2}{\sigma^2(\mathbf{x})} \qquad (8)$$

This squared distance function is positive, smooth, and (believed to be) unimodal -- allowing any of several local minimizers to be employed. (However, $A()$ is not defined outside the simplex, and explodes at the vertices, so care must be taken at the boundaries.)

## V. THE GROPE ALGORITHM

The search is initialized by probing $d$+1 points from the convex hull defining the space, or by absorbing previous results. (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 is sufficiently slight (7), iteratively:

1. Construct/Update the Delaunay triangulation, removing candidate probe locations for obsolete simplices from the ordered list.
2. For each <u>new</u> simplex $j$:
   a) Solve for $\mu_j(\mathbf{x})$ given vertices.
   b) Solve for $\sigma^2_j(\mathbf{x})$ given vertices and edges.
   c) Find the best probe location, $\mathbf{x}^*_j$, for the simplex by minimizing (8) (the squared, standardized distance to the goal, $A_j(\mathbf{x})$).
   d) Insert this candidate probe location into a list ordered by $A_j(\mathbf{x})$.
3. If locations on the intended convex hull remain unknown, probe there; otherwise, pop the head of the list, and probe at that location.

Steps 1 and 2c, the re-triangulation and the internal search of new simplices, are most affected by the number of probes, $N$, and problem dimension $d$. The added overhead is rather great (compared even to some other model-based searches). However, whenever probe computations are not trivial, that time should be more than compensated for by the algorithm's judicious choice of locations. (Time saved not "running" exceeds extra time spent "thinking"). For example, in [8] each probe for a guidance or 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!).

If $q$ processors are available (and if the application permits) $q$ - 1 probes may be removed from the head of the list and evaluated in step 3. The last processor could update the Delaunay triangulation, given the locations of the new probes, as the triangulation does not depend on their results, $\mathbf{y}$. Of course, the problem addressed should be reformulated to: find the best *set* of probes such that one is likely to exceed the goal. However, this simple q-at-a-time method should provide near-linear speedup for a common type of hard problem, where many regions of the domain must be explored (in which case the probes may as well be simultaneous).

## VI. EARLY EXPERIMENTAL RESULTS

A 2-dimensional prototype of GROPE has been prepared, using a recent program for planar Voronoi tessellation [19] (mapped into Delaunay triangles), and the *downhill simplex* method of function minimization [20] for the internal model search as programmed by [21]. The test function was the bimodal "Hosaki" equation [22]

$$(1 - 8x_1 + 7x_1^2 - \frac{7x_1^3}{3} + \frac{x_1^4}{4}) x_2^2 \exp(-x_2) \qquad (9)$$

illustrated in Fig. 7. The global minimum for $x_1 \in [0,5]$, $x_2 \in [0,6]$ is -2.345 at $\mathbf{x}$ = (4,2).

For testing purposes, nearness to the final answer was the stopping criterion. This is not usually possible for "field" applications, but allowed comparison with two random methods [22][23], and a promising recent model-based search [24] (Table 1).

Table 1: Hosaki 2-d Function Results

| Method | #Probes to Solution |
|---|---|
| Modified Random Creep [22] | **4 5 1** |
| Adaptive Random Search [23] | **8 3 0** |
| Sequential Design for Optimization [24] | **55** (constant param.)<br>**36** (linear param.) |
| GROPE (2nd trial) | **12** (goal = -3.0) |

The first GROPE run employed the known minimum as the goal, $y_g$, but crept too cautiously toward the final location, and failed to converge. The cautious approach to the minimum suggested that local probing was overly preferred to exploration of new areas; i.e., that the role of variance was too low relative to that of expectation. Accordingly, a more remote goal, $y_g$ = -3, was set (though, of course, still halting at $y_{min}$), leading to much improved results: only <u>12 probes</u>. The final triangulation of this second run is pictured in Fig. 8, where the vertices of the Delaunay triangulation are known probes, each "x" represents a candidate probe location for its

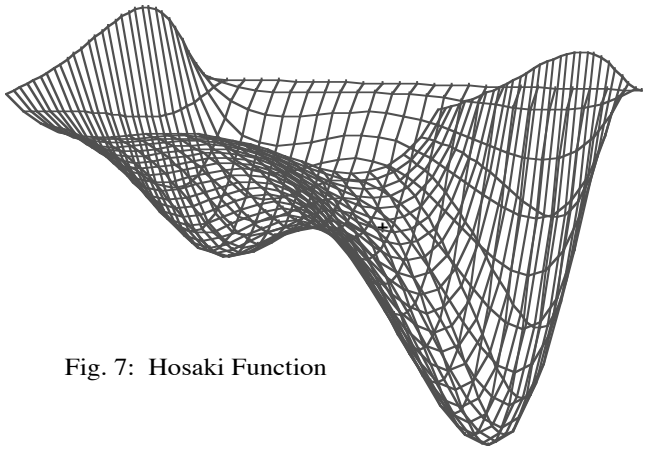


Fig. 7: Hosaki Function

triangle, and the "." denote discarded candidate locations (due to dissolution of the surrounding triangle), and "-", the position of the global minimum. The 11th probe value was -2.344. The number of triangles (and thus candidate locations) increases by two after each probe -- a property of the 2-d Delaunay triangulation.

## VII. POTENTIAL IMPROVEMENTS

Both GROPE runs in the example problem were initialized by probing the four corners of the search domain. Such rectangular bounding requires $2^d$ initial probes, which can be expensive in high dimensions. Furthermore, these first probes are taken in areas least expected to produce useful results: the domain boundaries. A minimum of $d+1$ probes (an initial simplex) can define the domain; yet, to roughly match the content of the hyper-rectangle, these probes would have to be even more extreme in location. It is convenient for the algorithm to have the bounds set initially, and always be performing a type of interpolation operation. But instead, perhaps some type of Bayesian technique, with distributions reflecting the desirability of probing in the center of the region, could be employed. This could restrain, to an adjustable degree, the otherwise linearly increasing variance beyond the outermost probe towards the bounds.

A simpler improvement, as suggested by the example application, would be to adjust the single current parameter of the algorithm ($y_g$) with time and/or performance -- a "relaxation" technique similar to other methods. Further research may show that a good goal scheduling strategy can be inferred for a problem from metrics of its ongoing results -- e.g., the distribution of probe results and its extreme, the (estimated) smoothness of the score surface and its variability, and the distribution of simplex content.

As evidenced by the first trial, regression singularities can occur when fitting the variance of long "thin" triangles near the convex hull having nearly collinear edges. Use of a robust fitting method (e.g. singular value decomposition with removal of small eigenvalues) is being investigated to remove this symptom of "overfit".

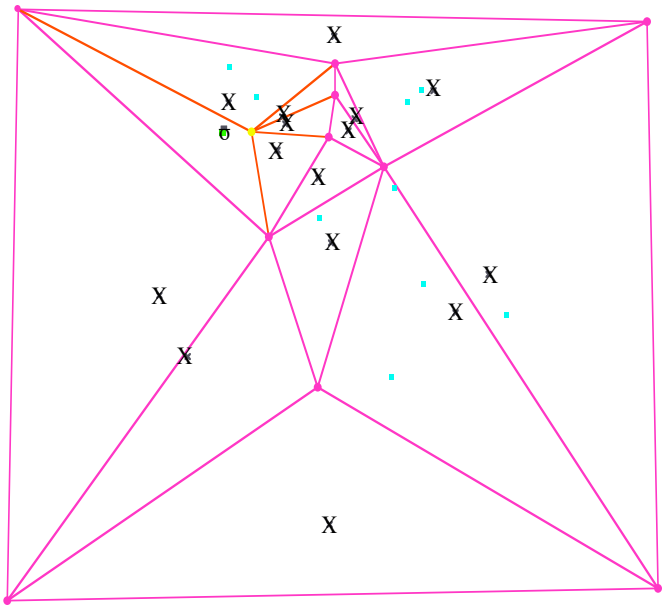


Fig. 8: Triangulation after near-Final (11th) Probe

## VIII. CONCLUSIONS

GROPE is a novel, efficient, model-based stochastic $R^d$ optimizer, in large part generalizing Kushner's elegant univariate method. For low- to medium-dimensioned problems (1-12 variables, say), GROPE should be more accurate in many fewer probes than conventional methods, and provide an interpretable confidence in the outcome.

## REFERENCES


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

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

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

[4] Stuckman, B.E. (1988). A Global Search Method for Optimizing Nonlinear Systems, *IEEE Trans. SMC* **18**, no. 6: 965-977.

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

[6] Perttunen, C.D. (1991). A Computational Geometric Approach to Feasible Region Division in Constrained Global Optimization, *Proc. IEEE SMC Conf.*, Charlottesville, Virginia, Oct. 13-16.

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

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

[9] 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.

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

[11] Adler, R.J. (1981). *The Geometry of Random Fields*. Wiley, NY.

[12] 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.

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

[14] Sibson, R. (1978). Locally Equiangular Triangulations, *Computer Journal* **21**, no. 3: 243-245.

[15] Joe, B. (1989). Three-Dimensional Triangulations from Local Transformations, *SIAM J. Sci. Stat. Com.* **7**: 514-539.

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

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

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

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

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

[21] Press, W.H., B.P. Flannery, S.A. Teukolsky, W.T. Vetterling (1988). *Numerical Recipes in C*. Cambridge Univ. Press, New York.

[22] Bekey, G.A., M.T. Ung (1974). A Comparative Evaluation of Two Global Search Algorithms, *IEEE Trans. SMC* **4**: 112-116.

[23] 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.

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