Stochastic Search

Griff L. Bilbro

Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
North Carolina State University

TR-94/1
January 1994

Stochastic Search

Griff L. Bilbro

ABSTRACT

Systems that select an optimal or nearly optimal member from a specified search set are reviewed with special emphasis on stochastic approaches such as simulated annealing, genetic algorithms, as well as other probabilistic heuristics. Because of local minima, selecting a global optimum may require time that increases exponentially in the problem size. Stochastic search provides advantages in robustness, generality, and simplicity over other approaches and is more efficient than exhaustive deterministic search.

INTRODUCTION

Optimization is the problem of searching a set S for an element $x \in S$ at which a given real objective function attains an extreme value. Without loss of generality we restrict our attention to minimization so that we seek $x \in S$ such that $f(x) \leq f(y)$ for all $y \in S$.

Optimization has provided a general and natural language for mathematically *formulating* problems in design or analysis ever since the invention of differential calculus. But except for the class of objective functions with a single minimum, actually *solving* the resultant mathematical formulation has too often defeated existing optimization techniques. The difficulty is due entirely to the intractable number of local minima that may occur in problems of high dimension; powerful methods have been developed for all other aspects of this problem[16, 41, 15].

We will address iterative algorithms that begin with an initial solution x_1 and repeatedly update the current solution x_i with some candidate y_i until termination. A deterministic optimizer generates candidates or accepts a

given candidate deterministically. A stochastic optimizer makes such choices probabilistically. Stochastic optimizers exist that converge with unit probability to the global minimum, but are faster than exhaustive search, simpler than heuristics, and sometimes provide the best known approach to difficult problems[27]. This report will discuss some of the advantages of stochastic search.

Until the advent of simulated annealing, there was no general approach to global optimization and the only alternatives were to reformulate the problem or to develop heuristics, both of which require expertise and time. Engineers and scientists had no generic, systematic approach to such problems: "It is a sobering thought that the only way to solve many engineering problems is still by trial and error"[1, originally quoted from *The Economist*].

Optimization is now becoming a generic industrial tool for engineering design as well as for the analysis and modeling of measured data. Part of this growing reliance on optimization is due to improved performance of computer hardware and software, part is due to more sophisticated initialization of local optimizers, but a significant part is due to the introduction in 1983 of simulated annealing, a generic *global* optimization strategy.

GLOBAL OPTIMIZATION

When S is finite, the problem is called a combinatorial optimization problem and there is at least one solution. The most important case when S is not finite is the case of continuous S, which is often called nonlinear optimization. When S is continuous, it is usually assumed to be compact in order to guarantee that a minimizer x exists if f is continuous.

The simpler case with finite S is often sufficient for practical applications in engineering. Even when an objective depends on continuous variables, the dependences that arise in image and signal processing, designing networks and filters, modeling measurements, and analyzing data are typically smooth enough that a minimum step size can be defined and used to *discretize* the problem, or to neglect the region of S between points on a discrete lattice. The domains of variables in engineering problems are typically bounded also, so that each variable often can be constrained to an interval.

Combinatorial optimization is then the problem of minimizing a function $f : X \rightarrow R$ where $X = \{1, ..., N\}$ is a finite set of feasible solutions. Unfortunately the number of states is too large to search exhaustively. Sometimes exhaustive search can be abbreviated without compromising the result. For other cases, efficient optimizers have been developed to exploit local structure to accelerate the search for a minimum. The use of such traditional optimizers is well justified, but they must sacrifice one or more of the three virtues: efficiency, generality, and guaranteed convergence. This category of optimizers includes the familiar deterministic techniques.

- Iterative descent, which is general but not robust. Descent algorithms for continuous problems are mature and powerful. These include most familiar generic nonlinear optimizers such as simple gradient based techniques, gradient techniques with variable metrics, and variants with low-storage requirements[15].

- Exhaustive search is "robust" but slow. The word robust is in quotes because the execution time required for exhaustive search is typically exponential in the number of variables. This time can grow so rapidly with problem size that such a problem is "intractable" in any practical sense. For example, a binary problem with 20 variables involves about 10^6 states or solutions to enumerate. A million solutions may be feasible, but 30 variables leads to 1000 times as large a search set. Unfortunately even small problems in image processing (*e.g.* restoration of an image with 128×128 pixels) may involve an astronomically large S, which is hopelessly intractable for any such straightforward approach.

$$\begin{aligned} 2^{20} &\approx 10^6, \\ 2^{30} &\approx 10^9 \\ 2^{128 \times 128} &\approx 10^{5000}, \end{aligned}$$

- Branch-and-bound or dynamic programming are rigorously characterizable and are helpful in special cases. These approaches are sometimes subtle because they may rely on sophisticated data structures or problem representations, and they require expertise to develop.

- Heuristics are sometimes the best possible approach to important and enduring problems. Heuristics often require profound understanding of

the problem structure and long development time, but may be superior to all other approaches when they exist. Heuristics are difficult to generalize because their reliance on problem specifics. Their performance is usually impossible to rigorously quantify when they contains "tricks" as is common.

NEW APPROACHES TO OPTIMIZATION

This second catagory of optimizers is the primary focus of this report because of they provide both generality and robustness with acceptable efficiency. These approaches are explicitly stochastic or are deterministic approximations to stochastic optimizers, such as

- The most powerful and promising of these stochastic search algorithms is Simulated Annealing (SA), the best known stochastic optimizer[29]. SA converts the optimization problem into a sampling problem and applies the powerful Metropolis sampling procedure[31]. SA was the first of its class to be proved correct[19]. Other approaches have been reported that exploit the powerful mathematics of statistical sampling theory[38, 2, 30, 28, 35]. Genetic algorithms are now being analyzed in this way[44].

- MFA and related techniques can be regarded as a deterministic approximations of SA[40, 39, 7, 12]. They depend on reversibly deforming the problem into an easier one as in the more rigorous but less general *Continuation Methods*, which are restricted to small problems[32]. Mean field annealing (MFA) is a deformation technique that produces good solutions quickly for much larger problems, such as those in image processing [24, 9, 14, 34]. MFA is related to Hopfield neural networks[25, 33, 5]. When MFA is inappropriate other deformation techniques sometimes can be developed[21, 17]. These techniques are of growing importance because they are versatile, fast, and are explicitly parallel, but unfortunately cannot be guaranteed to produce the globally optimal solutions.

Generic descent algorithm:

1. Select initial solution i.
2. Select candidate solution $j \in X_i$.
3. Replace $i \leftarrow j$ if and only if $f_j < f_i$.
4. If not done, go to 2.
5. Report i, f_i and stop.

Figure 1: The simplest iterative descent.

- Finally, there are *ad hoc* algorithms such as *multistart* algorithms which use a random number generator to repeatedly initialize a descent optimizer whose best result is kept. This approach is so simple it is often used in engineering, but its improvement over a single run is often disappointing for large problems.

All of these new approaches to optimization can be formulated as iterative descent optimizers by defining a neighborhood structure for X as some $X_i \subset X$ for each $i \in X$ where $j \in X_i$ implies $i \in X_j$. These algorithms iteratively replace a current solution i with neighboring solution $j \in X_i$. The neighborhood structure must permit the global minimum to be reached in a finite number of such steps from any initial solution as in the generic descent algorithm of Figure 1.

The performance of the algorithm of Figure 1 depends on the neighborhood structure. In the extreme, if $X_i = S$ for all i and the "Select" of step 2 takes the smallest element of X_i, then the algorithm arrives at a global minimum in the second iteration, but the "Select" is as hard as the original problem.

Many practical algorithms have been reported that employ smaller X_i.

Such a local algorithm may take the neighbors X_i of a binary vector x_i as the set that differ from x_i in exactly one bit. For continuous problems that have been quantized to discrete sites, a simple neighborhood of some point x_i could be defined by perturbing x_i to some y_i one that is a single quantization step away in a cardinal direction. The "done" condition of step 4 then holds when no neighbor is less than the current solution, a local minimum $f_i \leq f_j$ for all $j \in X_i$. Since $X_i \neq S$, there is no guarantee that f_i is the global minimum. However local methods are so simple they are widely used in spite of their dependence on initial conditions.

Most nonlinear optimizers are local and deterministic. For continuous problems, optimizers such as gradient descent, Newton's method, Fletcher-Reeves, and Levenberg-Marquardt are local optimizers that iteratively select candidates in some locally optimal way. These algorithms have the form of the generic descent algorithm of Figure 1. Such local deterministic methods are fast but cannot guarantee a global minimizer[15].

Except in special cases, deterministic codes are not reliable even in a relaxed sense: Not only do they fail to find the global optimum unless started from a good initial solution, but also they may fail to find even a good optimum. In some Bayesian image processing applications, local minima are so numerous that descent algorithms terminate at final solutions or images that are visually indistinguishable from the initial solutions or images.

It is easy to construct a problem that will defeat the generic descent algorithm of Figure 1: Allow the algorithm to operate from some initial point to some termination. Record the path of function evaluations that it makes as it generates a path from the initial point to the terminal point. We can assume that the path is not exhaustive since $X_i = S$, $\forall i$ is not practical for large S. We assume that the algorithm is deterministic also. We modify the objective by introducing a global minimum at any point not on the path of evaluated points in S recorded during the first execution. The algorithm will then fail for this modified objective. since no point on its previous path has been modified it will not deviate from that previous path. It will detect no difference from the original problem and will miss the global minimum that we artificially introduced. In practice, problems in high dimension may contain many local minima naturally.

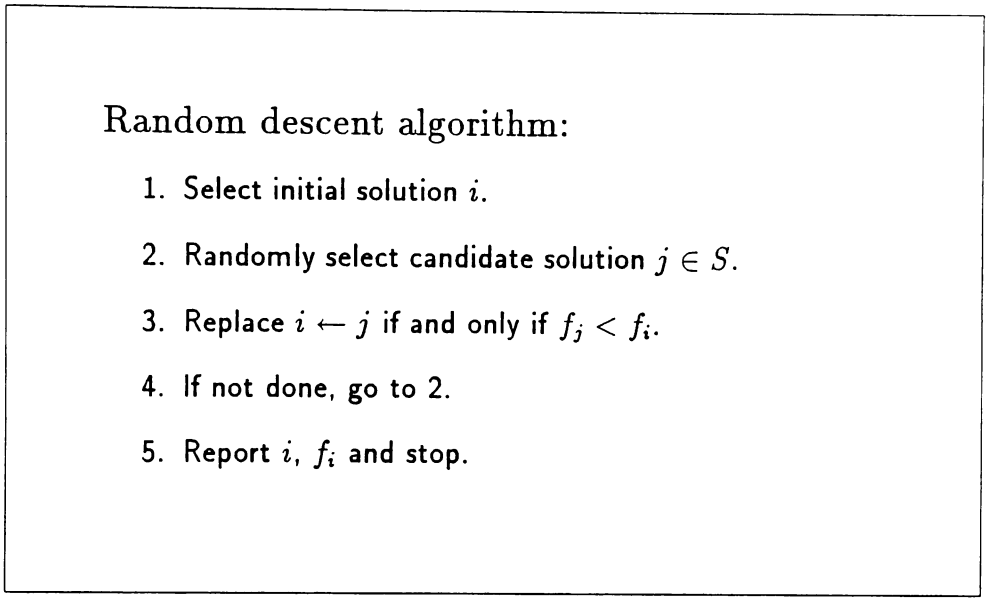

Figure 2: Random iterative descent.

RANDOM ALGORITHMS, SAMPLING, AND THE METROPOLIS PROCEDURE

Randomness can be used to recover the robustness of a descent algorithm at the expense of efficiency. In the crudest form, this becomes random exhaustive search as in Figure 2 in which the neighborhood is the entire search set S. This algorithm will eventually produce the global optimum for finite S, since eventually every member of S will be visited. Its finite-time performance is problem dependent, but is usually too poor to be useful. Multistart is obtained from random descent by replacing the word "solution" in step 2 with the word "initialization" and replacing f_i and f_j with the results of descending from i and j. Multistart therefore is provably convergent for finite S, but satisfactory only when there are few minima as in low dimensional problems.

The Metropolis procedure[31] is a rejection technique[28] which was originally developed to efficiently sample distributions in high dimensional spaces. It is powerful, general, and simple as is evident in Figure 3 which assumes only that the π is positive and has a finite integral. The Metropolis procedure can be proved to sample an arbitrary distribution; in the following section

Metropolis Procedure for distribution π:

1. Select initial solution i.
2. Select candidate solution $j \in X_i$ according to a uniform distribution over X_i.
3. Replace $i \leftarrow j$ if $\pi_j < \pi_i$. Otherwise compute

$$Q = \frac{\pi_j}{\pi_i}$$

   and replace $i \leftarrow j$ anyway with probability Q.
4. If not done, report i and go to 2.

Figure 3: The Metropolis Sampling Procedure produces as sequence of $i \in S$ which are distributed according to π.

we will prove this for a slightly more complicated case.

The Metropolis procedure is the basis for simulated annealing and other global optimization algorithms both for discrete applications. We will be principally concerned with discrete optimization, but randomness has been combined with descent to recover robustness such as Solis and Wets, Matyas, and Baba[38, 30, 2] who all treat continuous problems. Some approaches to continuous optimization use thermostatistical analogies such as the diffusion approach of Geman and Hwang[20] or the variant of simulated annealing proposed by Vanderbilt and Louie[43].

SIMULATED ANNEALING

In Simulated Annealing, the preceding Metropolis sampling procedure is run as the temperature is gradually reduced from an initially high value to a final low value. The basic SA algorithms is presented in Figure 4. In practice, SA

Simulated Annealing Algorithm:

1. Select initial solution i and high initial temperature T.
2. Randomly select candidate solution $j \in X_i$ from a uniform distribution over X_i.
3. Replace $i \leftarrow j$ if $f_j < f_i$. Otherwise compute

$$Q = \frac{e^{-f_j/T}}{e^{-f_i/T}}$$

   and replace $i \leftarrow j$ anyway with probability Q.
4. If not done, reduce T and go to 2.
5. Report i, f_i and stop.

Figure 4: Simulated annealing is a probabilistic descent algorithm that is only slightly more complicated than the generic descent algorithm. Note that $0 < Q < 1$ and therefore can be treated as a probability.

succeeds where deterministic codes fail. In theory, Simulated Annealing can be proved to converge to the global optimum under certain conditions.

Simulated annealing is not deterministic, but randomization alone does not guarantee global minimization. Consider a randomized descent algorithm which selects candidate j given current i from generator probabilities g_{ji}, and replaces i with j according to acceptance probabilities a_{ji}. This simple random descent algorithm with uniform $g_{ji} = 1/|X_i|$ and greedy $a_{ji} = 1$, if $f_j < f_i$ and $a_{ji} = 0$ otherwise, can also fail if f has local minima that are not neighbors.

BIASED METROPOLIS SAMPLING PROCEDURE

Biased Metropolis Procedure for sampling π:

1. Select initial solution i.
2. Select candidate solution $j \in X_i$ with probability $\gamma(j|i)$ given the current state is i.
3. Replace $i \leftarrow j$ if $\pi_j < \pi_i$. Otherwise compute

$$Q = \frac{\gamma(i|j)\pi_j}{\gamma(j|i)\pi_i}$$

   and replace $i \leftarrow j$ anyway with probability Q.
4. If not done, report i and go to 2.

Figure 5: Biased Metropolis Sampling Procedure produces as sequence of $i \in S$ which are distributed according to π. If generator $\gamma(i|j)$ is close to $\pi(i)$ then the procedure converges quickly. Note that $0 < Q < 1$ and therefore can be treated as a probability.

Simulated annealing based on the preceding Metropolis procedure is robust and is faster than exhaustive search but is still too slow to be convenient in many applications. Descent algorithms are fast but not robust because they exploit local structure which concentrates the search in locally optimal regions. Local structure can be used to advantageously bias the Metropolis sampling procedure without compromising convergence if the bias is carefully compensated[23]. The resulting sampling procedure will be used to construct a biased simulated annealing algorithm which is provably correct[1]. Experimental work has shown that this approach has useful practical application[6, 37, 8].

It is possible to prove that the Metropolis procedure and the biased vesion of the Metropolis procedure converges to the desired distribution. The first step of such a proof is to show that the target distribution is a stationary state or vector of the procedure. This implies that if the procedure ever arrives at the target distribution, it will remain there. We present the proof for this first result because it is easy and because it provides some insight.

Let the form of the Markov transition matrix from state x to state y be written as the product

$$P(y|x) = \gamma(y|x)a(y|x), \tag{1}$$

where $\gamma(y|x)$ is the probability of generating state y as a candidate given x as the current state, and $a(y|x)$ is the probability of accepting y as the next state given x as the current state. The goal is to use P to sample the Gibbs distribution π and for this π must be a stationary state of P. Given γ, it will suffice to construct P so that the *detailed balance condition* is satisfied

$$P(y|x)\pi(x) = P(x|y)\pi(y). \tag{2}$$

Combining Equations 1 and 2 yields a condition on a

$$\frac{\pi(x)\gamma(y|x)a(y|x)}{\pi(y)\gamma(x|y)a(x|y)} = 1. \tag{3}$$

The probability a can be chosen

$$a(y|x) = \min(1, q(y|x)) \tag{4}$$

for some q which can be chosen to obey

$$q(y|x) = \frac{1}{q(x|y)}. \tag{5}$$

This allows the separation of Equation 3 into three cases $q(y|x) > 1$, $q(y|x) = 1$, and $q(y|x) < 1$ to obtain

$$q(y|x) = \frac{\gamma(x|y)\pi(y)}{\gamma(y|x)\pi(x)}. \tag{6}$$

The corresponding algorithm is displayed in Figure 5. The Markov chain resulting from the formal algorithm Figure 5 or equivalently from the transition matrix of Equation 6 informally constructed in the preceding paragraph therefore has the following desirable behavior, first reported by Hastings[23] who obtained the result in a different way.

Theorem 1 (Hastings[23]): The distribution π of Equation 6 is a stationary distribution of the transition matrix

$$P(y|x) = \begin{cases} \gamma(y|x)\min(1, q(y|x)) & \text{if } y \neq x \\ 1 - \sum_{z \neq x} P(z|x) & \text{if } y = x \end{cases}, \tag{7}$$

where q is given by Equation 6

$$q(y|x) = \frac{\gamma(x|y)\pi(y)}{\gamma(y|x)\pi(x)}.$$

and where γ is a positive definite but otherwise arbitrary distribution.

Note that only the ratio of π's appear in Equation 6, so that the normalization Z is not necessary for the procedure, just as in the usual Metropolis procedure. Moreover the Markov chain will actually converge to the Gibbs' distribution, as is shown in the following result, also claimed by Hastings[23].

Theorem 2: The finite homogeneous Markov chain generated by transition matrix of Equation 7 converges asymptotically to $\pi(x)$.

Theorem 1 shows that Hastings' procedure for fixed P has the correct stationary distribution independently of the particular generator used in the procedure. Theorem 2 shows that the proposed procedure actually converges to this Gibbs distribution (in infinite time). This is sufficient to show convergence of a "homogeneous simulated annealing algorithm" which involves an infinite homogeneous Markov chain at each of a sequence of temperatures decreasing to zero[42].

Theorem 3: Let

$$\pi(x) = \frac{1}{Z}\exp\left(-\frac{U(x)}{T}\right) \tag{8}$$

where Z is a normalization factor which depends only on the temperature T. Then the usual homogeneous annealing algorithm[42] based on Equation 7 converges to a global minimum of U with probability 1.

The homogeneous "algorithm" cannot be exactly implemented since it requires an infinite number of transitions at each temperature, but the preceding homogeneous algorithm can be modified to converge to the global minimum of U with probability 1 with a logarithmic cooling schedule like that of Geman and Geman[19]. An annealing algorithm is called inhomogeneous when the associated Markov chain is inhomogeneous (or nonstationary) due to changes in temperature (and therefore the transition matrix) after a finite number of transitions. One procedure is to first produce an inhomogeneous Markov chain that is *weakly ergodic* for some large enough constant C.

Theorem 4: The inhomogeneous Markov chain for the transition matrix of Theorem 1 with π given by Equation 8 is weakly ergodic for fixed γ and cooling schedule given by

$$T_k = \frac{C}{\ln(1+k)} \tag{9}$$

so that it is asymptotically independent of the initial state, then prove that the chain is *strongly ergodic*, and finally that it asymptotically converges to the global minimum of U[26, 19, 22].

Simulated annealing guarantees global minimization by carefully choosing $a_{ij} > 0$ for $f_j > f_i$ with a Metropolis procedure or a Gibbs sampler[19]. A typical scheme employs uniform $g_{ij} = 1/|X_i|$ and Metropolis $a_{ij} = \min(1, \exp(f_i - f_j)/T_k)$ in the k^{th} step where $T_k = C/\ln(1+k)$ for large k. Simulated annealing is fastest when C and $|X_i|$ are small. In practice, the premises of convergence are often violated to make SA run faster.

Simulated Annealing is faster than exhaustive search when good solutions occur in regions of slightly poorer solutions[36, 6] as in Figure 6 of a Rastrigin-like function[41]. Simulated Annealing algorithms are particularly efficient on functions with smooth global structure perturbed by smaller local variations. SA is not well suited to minimizing functions with deep, isolated minima ("golf-course" or "gopher-hole" functions) such as the Shekel-like function of Figure 7[41].

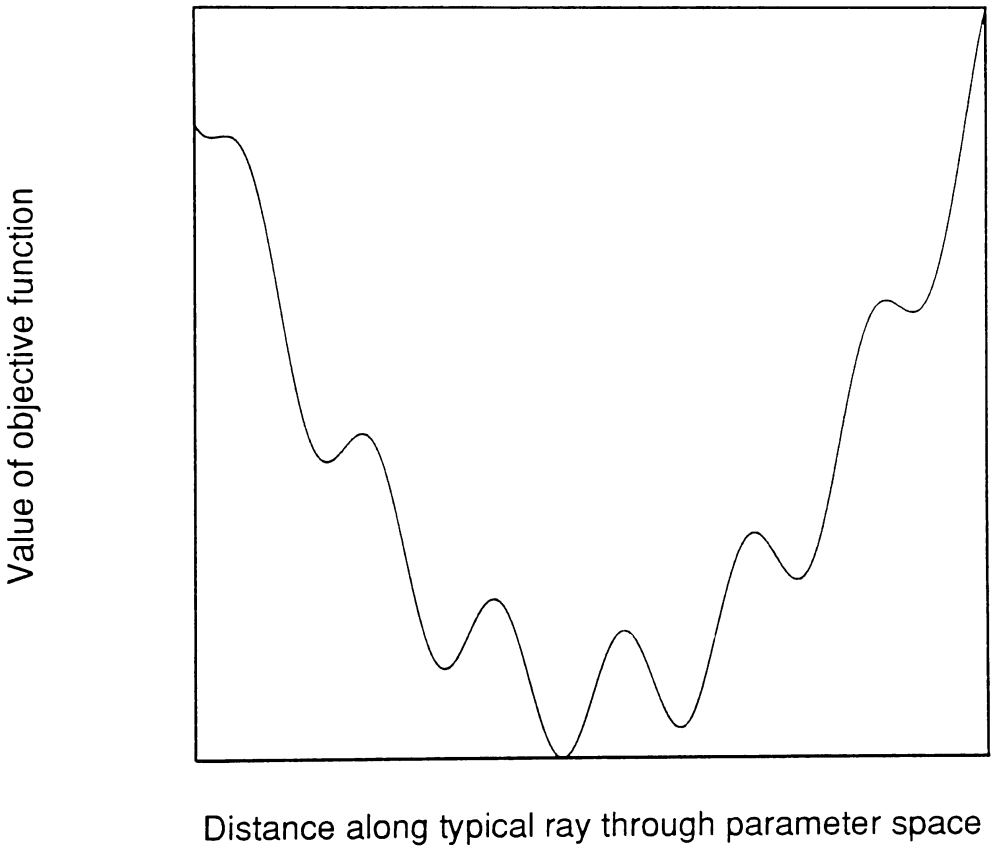

Figure 6: A Good SA problem. One dimensional slice through a Rastrigin-like function.

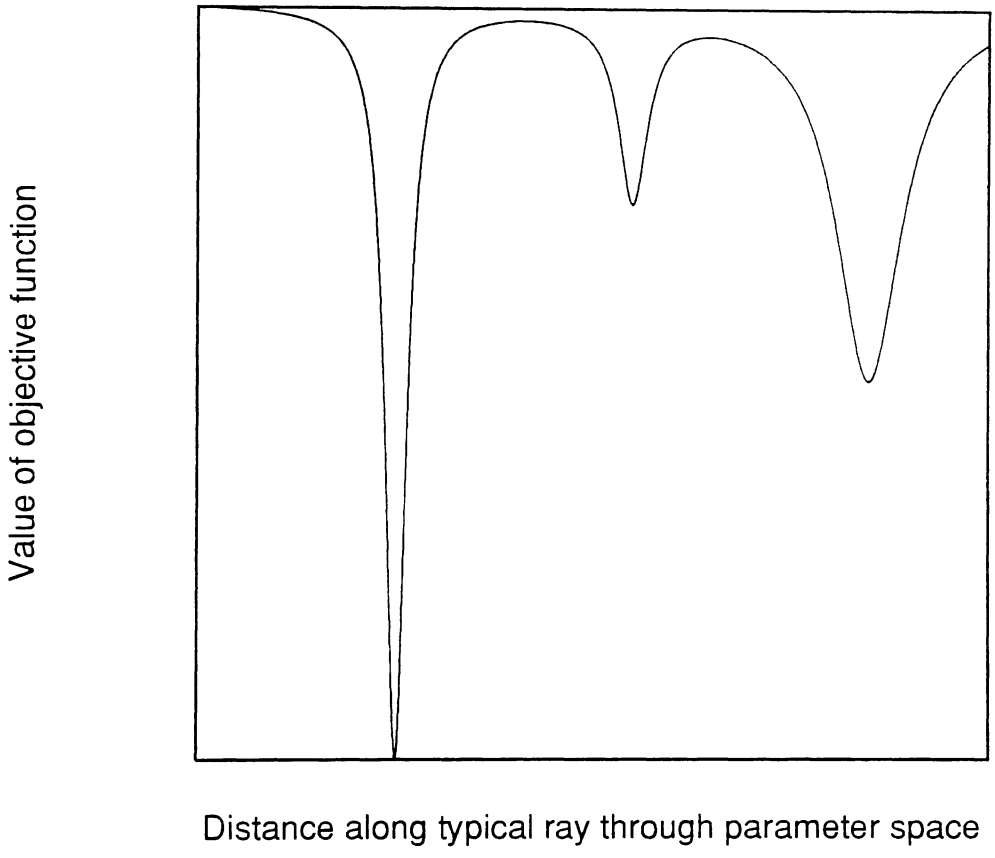

Figure 7: A bad SA problem. One dimensional slice through a Shekel-like function.

BIASED SIMULATED ANNEALING

Simulated annealing can be accelerated by biasing the g_{ij} to concentrate the search in promising regions, but this bias will cause erroneous convergence unless the a_{ij} are adjusted to compensate[1]. One early approach used a biased generator without modification of the acceptance probabilities to obtain promising but erratic results: the covariance matrix of the Markov chain was adaptively estimated and used to orient the search in anisotropic search spaces. That approach discarded the scaling information contained in the covariance matrix and it required complicated heuristics to prevent premature convergence[43]. In a later image restoration, a non-adaptive biased generator was shown to reduce run times without compromising theoretical convergence. In that application, the Bayesian *maximum a posteriori* objective contained a likelihood part suitable for constructing a fixed generator appropriate to the problem[3]. More recently, when good g_{ij} are not available *a priori*, they have been adaptively accumulated in a binary tree but without guaranteed results[6, 8].

The benefit of biasing the generator can be demonstrated experimentally. Since the generator is Gaussian, it is approximately uniform in a region of width s around the observed image g and zero elsewhere. For $s >> \sigma$, the difference between g and f is insignificant because it is of order σ. In that regime, the generator is indistinguishable from a uniform perturbative generator around the current state f, so that the behavior of the usual Metropolis-based simulated annealing algorithm with uniform generator can be estimated. The dependence of total restoration time versus s is plotted in Figure 10 for the polygonal foreground of value 10 above a background of value 0 degraded by additive Gaussian noise of zero mean and standard deviation 3, as shown in Figure 8. The upper curve (labeled "max") in Figure 10 shows the time to reliably obtain a restoration that differs from the original image by less than 1 gray level except for at most 1 percent of the pixels and corresponds to the restoration shown in Figure 9. The lower curve (labeled "min") shows the time to obtain restorations with 98 percent of the pixels within 2 gray levels of the original, requiring 1 second at $s = \sigma = 3$. For either curve in Figure 10, the optimal value of s is practically equal to the $\sigma = 3$ of the noise added to the original image. At $s = 128$, the generator is about as wide as in a Metropolis based simulated annealing algorithm with uniform generator over the state space $(0, 1, 2, ..., 255)$. Figure 10 indicates

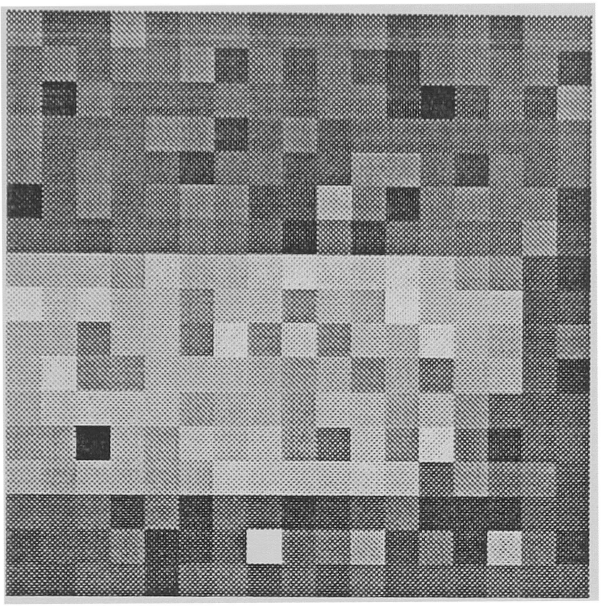

Figure 8: Data image degraded by noise.

that at $s = 100$, a uniform Metropolis algorithm runs about 50 times longer than at $s = \sigma$.

Figure 11 is a 64x64 cartoon image taken with a conventional video camera and frame grabber. The original image has a minimum value of 35, a maximum of 252, a mean of 42, and a standard deviation of 44. It was corrupted in Figure 12 with zero mean Gaussian noise of standard deviation 10 and restored in Figure 13.

DETERMINISTIC APPROXIMATIONS TO SA

MFA algorithms resemble the Graduated Nonconvexity Algorithm (GNC) developed by Blake and Zisserman[11] to solve certain image processing problems. MFA is based on Simulated Annealing (SA) and derives its power and generality from that popular optimization procedure. Geiger and Girosi first reported a relation between MFA[18].

MFA differs from SA by analytically approximating the relevant Gibbs distribution rather than stochastically simulating it. Mean field theory pro-

Figure 11 is a 64x64 cartoon image taken with a conventional video camera and frame grabber. The original image has a minimum value of 35, a

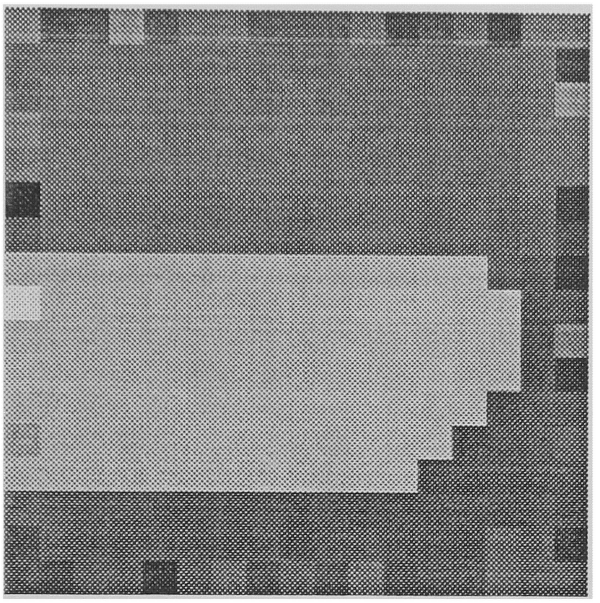

Figure 9: Restoration of data image. Note that the edge pixels are not varied by this algorithm.

vides a deterministic approximation to a Gibbs distribution which also can be cooled in the same way to produce a Mean Field Annealing (MFA) algorithm. Many SA algorithms can be converted to analogous MFA algorithms that run in 1/50 the time required by the SA version[40, 5, 24, 7]. However because it is an approximation, MFA does not inherit any guarantee of convergence even when the analogous SA does converge.

The mean field is usually restricted to Ising-like systems described by an energy function (or Hamiltonian) involving binary variables $s = \{s_i\}_{i=1}^{i=N}$ but recently has extended MFA to a wider class of problems[10, 9].

GENETIC ALGORITHMS

A popular new approach to global optimization in high dimensional spaces are Genetic Algorithms[13]. The basic Genetic Algorithm (GA) has a similar form to the simulated annealing algorithms. Figure 14 defines a basic genetic algorithm.

The most salient difference between SA-type algorithms and GA-type algorithms is the *population* of solutions used by GA. However SA is not

vides a deterministic approximation to a Gibbs distribution which also can be cooled in the same way to produce a Mean Field Annealing (MFA) algorithm. Many SA algorithms can be converted to analogous MFA algorithms

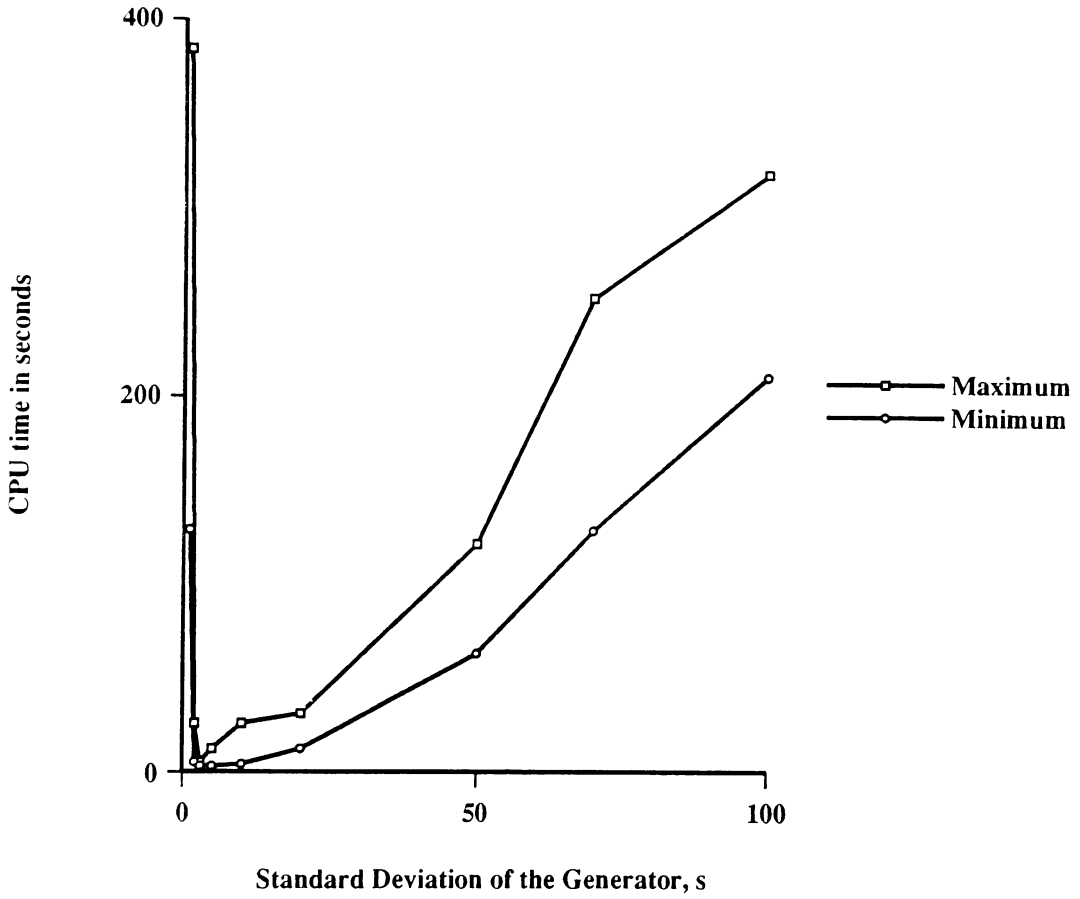

Figure 10: A plot of restoration time in seconds versus the standard deviation s of the generator. Fastest restoration occurs near $s = \sigma$. Restoration quality is somewhat subjective and the two curves attempt to bracket the time required for good restorations.

Figure 11: Original 64x64 cartoon video image.

Figure 12: Figure 11 plus Gaussian noise.

Figure 13: Restoration of Figure 12.

fundamentally limited to a single solution and can be formulated for a population like GA with apparently the same advantage[4]. A more fundamental difference between GA and SA is that the cross or mating operation of GA are not typically reversible. This irreversibility excludes much of the thermostatistical reasoning that is useful in proving convergence in SA.

CONCLUSIONS

Stochastic global optimizers, such as simulated annealing, are simple to use, provably robust, more efficient than exhaustive search, and more general than special purpose heuristics. Recent advances in biased versions of Simulated Annealing have reduced execution times by two orders of magnitude.

References

[1] S. Anily and A. Federgruen. Simulated annealing methods with general acceptance probabilities. *J. Appl. Prob.*, 24:657–667, 1987.

lation like GA with apparently the same advantage[4]. A more fundamental difference between GA and SA is that the cross or mating operation of GA are not typically reversible. This irreversibility excludes much of the ther-

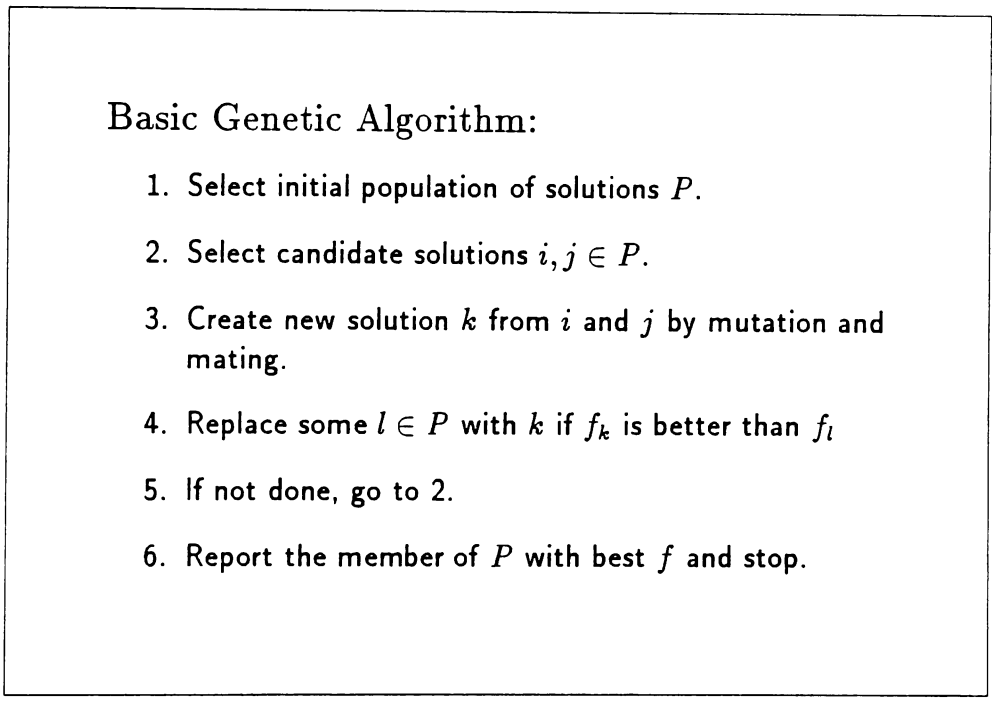

Figure 14: The basic Genetic Algorithm stochastically optimizes a population of solutions.

[2] J. Baba. A new approach for finding the glboal minimum of error function of neural networks. *Neural networks*, pages 367–373, 1989.

[3] G. L. Bilbro. A general method for accelerating SA for Bayesian restoration. In S.-S. Chen, editor, *SPIE Conference on Stochastic Methods in Sig. Proc, Image Proc., and Computer vision*, pages 88–98. SPIE, San Diego, CA, 1991. Invited paper.

[4] G. L. Bilbro, Lester C. Hall, and L. A. Ray. A provably convergent inhomogeneous genetic annealing algorithm. In S.-S. Chen, editor, *SPIE Vol. 1766 Neural and Stochastic Methods in Image and Signal Processing*. SPIE, San Diego, CA, 1992. Invited paper.

[5] G. L. Bilbro, T. K. Miller, W. E. Snyder and D. E. Van den Bout, M. W. White, and R. C. Mann. Optimization by the Mean Field Approximation. In *Advances in Neural Network Information Processing Systems 1*, pages 91–98. Morgan-Kauffman, San Mateo, 1989. Reprinted from 1988 NIPS, Denver, CO.

[6] G. L. Bilbro, M. B. Steer, R. J. Trew, C.-R. Chang, and S. G. Skaggs. Extraction of the parameters of equivalent circuits of microwave transistors using tree annealing. *IEEE Trans. Microwave Theory and Techniques*, Nov. 1990.

[7] Griff L. Bilbro and Wesley E. Snyder. Mean field annealing: An application to image noise removal. *Journal of Neural Network Computing*, Fall:5–17, 1990.

[8] Griff L. Bilbro and Wesley E. Snyder. Optimization of functions with many minima. *IEEE Trans. Systems, Man, and Cybernetics*, 21(4), 1991.

[9] Griff L. Bilbro, Wesley E. Snyder, Steven J. Garnier, and James W. Gault. Mean field annealing: A formalism for constructing GNC-like algorithms. *IEEE Transactions on Neural Networks*, 3(1), 1992.

[10] Griff L. Bilbro, Wesley E. Snyder, and Reinhold C. Mann. The mean field minimizes relative entropy. *Journal of the Optical Society of America A*, 8(2):290–294, February 1991.

[11] A. Blake and A. Zisserman. *Visual Reconstruction.* The MIT Press, Cambridge, Mass., 1987.

[12] P. C. Doerschuk C. H. Wu. Cluster approximations for statistical image processing. In S.-S. Chen, editor, *SPIE Vol. 2032 Neural and Stochastic Methods in Image and Signal Processing.* SPIE, San Diego, CA, 1993. Invited paper.

[13] L. Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1991.

[14] D. E. Van den Bout and T. K. Miller. Graph partitioning using annealed neural networks. *IEEE Trans. Neural Networks*, 1(2):192–203, 1990.

[15] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall Series in Computational Mathematics, 1983.

[16] L. C. W. Dixon and G. P. Szegő, editors. *Towards Global Optimization 2.* North Holland, Amsterdam, 1978.

[17] Steven J. Garnier, Griff L. Bilbro, James W. Gault, and Wesley E. Snyder. Magnetic resonance image restoration. *Journal of Mathematical Imaging and Vision*, 1993. Accepted for publication.

[18] D. Geiger and F. Girosi. Coupled markov random fields and mean field theory. In D. S. Touretzky, editor, *Advances in Neural Network Information Processing Systems 2.* Morgan-Kauffman, San Mateo, 1990.

[19] D. Geman and S. Geman. Stochastic relaxation, Gibbs Distributions, and the Bayesian restoration of images. *IEEE Transactions on PAMI*, PAMI-6(6):721–741, November 1984.

[20] S. Geman and C. R. Hwang. Diffusions for Global Optimisation. *SIAM Journal of Control and Optimisation*, 24(5):1031–1043, September 1986.

[21] Bennett R. Groshong, Griff L. Bilbro, and Wesley E. Snyder. Eddy current flaw image restoration by constrained gradient descent. *Journal of nondestructive evaluation*, 10(4), 1991.

[22] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2):311–329, 1988.

[23] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[24] H. P. Hiriyannaiah, G. L. Bilbro, W. E. Snyder, and R. C. Mann. Restoration of piecewise constant images via mean field annealing. *Journal of the Optical Society of America A*, pages 1901–1912, December 1989.

[25] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

[26] D. W. Isaacson and R. W. Madsen. *Markov Chains, theory and applications.* Wiley, New York, 1976.

[27] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: an experiental evaluation, parts i and ii. Technical report, AT&T Bell Laboratories, 1987.

[28] M.H. Kalos and P. A. Whitlock. *Monte Carlo Methods, Volume 1: Basics.* Wiley and Sons, New York, 1986.

[29] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[30] J. Matyas. Random optimization. *Automation and Remote Control*, pages 246–253, 1965.

[31] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Chem. Physics*, 21:1087–1092, 1953.

[32] A. Morgan. *Solving polynomial systems using continuation for engineering and scientific problems.* Prentice-Hall, 1987.

[33] C. Peterson and J.R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1(5):995–1019, 1987.

[34] C. Peterson and J.R. Anderson. Neural networks and np-complete optimization problems; a peformance study on the graph bisection problem. Technical Report MCC-EI-287-87, Available from the Microelectronics and Computer Technology Corporation, 1987.

[35] R. Y. Rubinstein. *Monte Carlo Optimization, Simulation and Sensitivity of Queueing Networks.* John Wiley & Sons, 1986.

[36] R. Rutenbar. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine*, 5(1):19–26, 1989.

[37] Wesley E. Snyder, Griff L. Bilbro, Ambalavaner Logenthiran, and Sarah A. Rajala. Optimal thresholding a new approach. *Pattern Recognition Letters*, 11(12):803–810, December 1990.

[38] F. J. Solis and F. B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, pages 19–30, 1981.

[39] C. M. Soukoulis, G. S. Grest, and K. Levin. Irreversibility and metastability in spin-glasses. II. heisenberg model. *Phys. Rev. B*, 28(3):1510–1523, August 1983.

[40] C. M. Soukoulis, K. Levin, and G. S. Grest. Irreversibility and metastability in spin-glasses. I. ising model. *Phys. Rev. B*, 28(3):1495–1509, August 1983.

[41] A. Törn and A. Žilinskas. *Global Optimization.* Lecture Notes in Computer Science. Springer-Verlag, 1989.

[42] P.J.M Van Laarhoven and E. H. Aarts. *Simulated Annealing: Theory and Applications.* Reidel, Dordrecht, Holland, 1987.

[43] D. Vanderbilt and S. G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 36:259–271, 1984.

[44] M. D. Vose. Closer look at mutation in genetic algorithms. In S.-S. Chen, editor, *SPIE Vol. 2032 Neural and Stochastic Methods in Image and Signal Processing.* SPIE, San Diego, CA, 1993. Invited paper.