SPARSE NETWORK ARRAY PROCESSING EMPLOYING PRIOR COVARIANCE KNOWLEDGE

Edward J. Baranoski
MIT Lincoln Laboratory, Rm. J-118D, 244 Wood Street, Lexington, MA 02173
E-mail: ejb@ll.mit.edu

ABSTRACT

This paper examines an alternative multi-dimensional adaptive array processing architecture which provides a unique highly-parallelisable algorithm suitable for distributed processing. The principle is to perform interference cancellation on each element using a different sparse sampling of the remaining elements as auxiliary inputs. By doing so, special correlations in the data can be exploited to significantly reduce the degrees of freedom required in each adaptive process. This reduces both the computation count and the number of samples required for adaptivity. An example space-time adaptive nulling application of airborne clutter shows near optimal performance with a factor of four computational savings over equivalent space-time techniques.

1. INTRODUCTION

This paper addresses an alternative approach to adaptive array processing which exploits *a priori* knowledge of the interference scenario to achieve computational savings. Many early adaptive arrays were based upon LMS or Howells-Appelbaum adaptive processors [1] which offered adaptive performance at relatively low cost and complexity. Such techniques often have problems with the amount of data required before the weight vectors converge to their proper values. Faster converging algorithms have started becoming practical as digital devices have matured. Sample matrix inversion (SMI) adaptive algorithms are now considered practical and are used for many applications when processing blocks of data. However, the cost of SMI algorithms becomes prohibitive as the array dimensionality grows. An example of this is pulsed radar applications for which space-time adaptive processing [2, 3] are often employed. This family of adaptive algorithms uses both spatial and temporal degrees of freedom to null clutter using the clutter's spatio-temporal correlation. The degree of freedom requirement is now the product of the number of spatial elements and the number of pulses used. Since the computational requirements of an SMI-based processor increase as the square of the degrees of freedom, this can become prohibitive and limit the allowable size or temporal extent of such a problem.

Sparse arrays have been proposed as a method of obtaining the same resolution as a longer array at a fraction of the cost. However, the sidelobes of the resulting patterns were quite high, and the adaptive patterns were vulnerable to aliased nulls in the presence of certain interference environments [4]. The latter point was due to aliasing because of subsampling the array aperture. In [5], a similar sparse approach was employed for constrained arrays. However, that technique used a single sparse projection for the full array, and was therefore subject to the same limitations (e.g., poor sidelobes, vulnerability to certain interference scenarios due to aliasing) as conventional sparse arrays.

Our algorithm [6] breaks down large adaptive array processing tasks into many smaller, more manageable adaptive problems. It is based on the concept of element nulling. In principal, the ideal solution to an optimal adaptive weight vector

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{v} \tag{1}$$

for an N-element adaptive antenna array using a steering vector $\mathbf{v}$ and interference covariance matrix $\mathbf{R}$ can be reformulated as simultaneously "whitening" each element using all other elements as auxiliary inputs. This is similar to the sidelobe canceling approach of adaptive array processing [1]. Consider the element nulling weight vector

$$\mathbf{w}_n = \mathbf{R}^{-1}\mathbf{e}_n \tag{2}$$

where $\mathbf{e}_n$ is an N-element column vector consisting of a one in the n-th position and zeros elsewhere,

$$\mathbf{e}_n = [\ 0 \ \ \cdots \ \ 0 \ \ 1 \ \ 0 \ \ \cdots \ \ 0\ ]^T. \tag{3}$$

Note that $\mathbf{w}_n$ is simply the n-th column of $\mathbf{R}^{-1}$. The overall weight vector in (1) is equivalent to

$$\mathbf{w} = \sum_{n=1}^{N} \mathbf{w}_n v_n \tag{4}$$

where v_n is the n-th element of $\mathbf{v}$. This approach is obviously not as efficient as solving for the weight vector directly using (1). However, when the rank of the interference is significantly less than N, each $\mathbf{w}_n$ vector can be formed using only a small subset of the N elements. We have called this new technique Sparse Network Array Processing (SNAP) due to its use of sparsely sampled subsets of elements which can be fed through a network of small adaptive processors.

The sparse network approach is less susceptible to the deficiencies common with sparse arrays since the sparse network treats a fully populated array as N different overlapping sparse arrays. The nulling performance of each is maintained, but the sidelobe structure of each is uncorrelated and therefore will not appreciably degrade the overall sidelobe structure when considered together.

In [6], the sparse network approach was introduced and a space-time clutter nulling example was presented.

This work was supported in part by Air Force contract F19628-95-C-0002.

In this paper, we re-examine the selection criterion for forming the sparse nulling sets. In Section 2 we look at the theory of sparse networks and examine how *a priori* covariance information can be used to select the sparse element nulling sets. In Section 3, we show the implementation of this algorithm including sparse element selection.

2. THEORY

In this section we will address the theory behind the sparse network algorithm. To aid in our discussion, we will use pseudo-code notation to describe portions of matrices. Given the m-length vector $\mathbf{b}$ and the n-length vector $\mathbf{c}$, the notation $\mathbf{A}(\mathbf{b},\mathbf{c})$ denotes the $(m \times n)$ submatrix consisting of the intersection of the rows specified by the elements of $\mathbf{b}$ and the columns specified by the elements of $\mathbf{c}$. Similarly, the notation $\mathbf{a}(\mathbf{b})$ describes the m-element subvector consisting of the elements in $\mathbf{a}$ described by the m elements in $\mathbf{b}$. An asterisk '$*$' in either the row or column position specifies all rows or columns. The superscript 'H' denotes Hermitian or complex transpose of a vector or matrix.

SNAP attempts to perform element whitening using only a fraction of the remaining elements as auxiliary elements [6]. Let us denote the subset of D elements which will be used to null element n as the vector $\mathbf{p}_n$, consisting of the elements $p_n(1)$ through $p_n(D)$. Let the element selection matrix $\mathbf{E}_n$ be the matrix whose columns are the unit vectors specified by the elements in $\mathbf{p}_n$, such that

$$\mathbf{E}_n = [\ \mathbf{e}_{p_n(1)} \quad \cdots \quad \mathbf{e}_{p_n(D)}\ ]. \qquad (5)$$

Given the full N-element array data vector $\mathbf{x}$ with covariance $\mathbf{R}$, we can express the sparse subarray vector as $\mathbf{x}_n = \mathbf{E}_n^H \mathbf{x} = \mathbf{x}(\mathbf{p}_n)$ and its covariance as $\mathbf{R}_n = \mathbf{R}(\mathbf{p}_n, \mathbf{p}_n)$. The element nulling vector using this subarray is

$$\tilde{\mathbf{w}}_n = \mathbf{R}_n^{-1}\tilde{\mathbf{e}}_n, \qquad (6)$$

where $\tilde{\mathbf{e}}_n = \mathbf{E}_n^H \mathbf{e}_n = \mathbf{e}_n(\mathbf{p}_n)$. This can be applied to the entire array by applying each element of $\tilde{\mathbf{w}}_n$ to its appropriate element in the full matrix, yielding the sparse element weight vector[1]

$$\mathbf{w}_n = \mathbf{E}_n \tilde{\mathbf{w}}_n = \mathbf{E}_n \mathbf{R}_n^{-1} \mathbf{E}_n^H \mathbf{e}_n. \qquad (7)$$

Using the nulling matrix $\mathbf{W} = [\ \mathbf{w}_1 \quad \cdots \quad \mathbf{w}_N\ ]$. and (4),

$$\mathbf{w} = \sum_{n=1}^{N} \mathbf{w}_n v_n = \mathbf{W}\mathbf{v}. \qquad (8)$$

represents the final weight vector.

As a simple example, consider the $N \times N$ Toeplitz covariance matrix $\mathbf{R}$ whose (i,j) entry is $r(i,j) = 0.9^{(i-j)}$. This matrix is full rank with no clear separation between 'signal' and 'noise' eigenvalues, so subspace processing using dimensionality reduction is not possible [7]. However, the inverse of this $\mathbf{R}$ is tridiagonal (although not Toeplitz). A perfect nulling matrix can be formed using $D = 3$ auxiliary elements for each sparse weight vector. To null element n for $1 < n < N$, the elements $(n-1, n, n+1)$ are used to form $\mathbf{w}_n$, whereas the first three elements are used to null element $n = 1$ and the last three elements are used to null element $n = N$. Instead of the adaptive problem being on the order of $O(N^3)$ operations, the sparse network problem only requires $NO(D^3) = 27N$ operations, which demonstrates the potential benefit of sparse network approaches for certain covariance structures.

In order for the sparse network concept to work efficiently, the sparse selection sets $\mathbf{p}_n$ for all n should be chosen *a priori* based upon the expected structure of the covariance matrix. These pre-formed sparse element sets will then be used by N adaptive processors. Selecting the sparse element sets is not a trivial matter. While closed-form solution strategies do not exist, the following sections explain several principles which can guide an iterative search to find good (although not necessarily optimal) sparse element sets. Each section looks at the sparse nulling procedure at a progressively deeper level, starting with individual element nulling and finishing with the overall system nulling performance.

2.1. Element-level SINR

This section outlines how to select the sparse element set for each element independently of the remaining elements. Each sparse weight vector is responsible for nulling the interference for the element for which the weight vector was formed. To judge the effectiveness of this nulling, we can compute the signal to interference plus noise ratio (SINR) S_n for the sparse element weight vector $\mathbf{w}_n$ for the element weight vector $\mathbf{e}_n$ as

$$S_n = \frac{\left|\tilde{\mathbf{w}}_n^H \tilde{\mathbf{e}}_n\right|^2}{\tilde{\mathbf{w}}_n^H \mathbf{R}_n \tilde{\mathbf{w}}_n} = \tilde{\mathbf{e}}_n^H \mathbf{R}_n^{-1} \tilde{\mathbf{e}}_n = \mathbf{w}_n^H \mathbf{e}_n, \qquad (9)$$

which is simply the diagonal element of $\mathbf{R}_n^{-1}$ corresponding to element n. This provides some insight into which elements to select to form the sparse element weight vector. Since the trace of $\mathbf{R}_n^{-1}$ is simply $\sum_{i=1}^{D} \lambda_i^{-1}$ where λ_i is the i-th eigenvalue of $\mathbf{R}_n$, S_n is upper bounded by λ_{min}^{-1}. This implies that the auxiliary elements to null element n must be chosen so that there is at least one noise eigenvalue in the resultant sparse covariance $\mathbf{R}_n$. Ordinarily, this would imply that the number D of auxiliary elements used exceeds the number of significant eigenvalues (e.g., the number of sources) present in the original covariance matrix $\mathbf{R}$. However, the structure of the original covariance might allow significantly fewer auxiliary elements.

This is only one facet of SINR, since the signal power must be preserved. Little is gained if the interference is nulled at the expense of severely degrading the signal power. Therefore, we must consider the performance of the sparse weight vector $\mathbf{w}_n$ as a function of the steering vector $\mathbf{v}(\theta)$, which is

$$S_n(\theta) = \frac{|\tilde{\mathbf{w}}_n^H \mathbf{E}_n^H \mathbf{v}(\theta)|^2}{\tilde{\mathbf{w}}_n^H \mathbf{R}_n \tilde{\mathbf{w}}_n} = \frac{|\mathbf{w}_n^H \mathbf{v}_\theta|^2}{\mathbf{w}_n^H \mathbf{e}_n} \qquad (10)$$

since

$$\mathbf{w}_n^H \mathbf{R} \mathbf{w}_n = \tilde{\mathbf{w}}_n^H \left(\mathbf{E}_n^H \mathbf{R} \mathbf{E}_n\right) \tilde{\mathbf{w}}_n = \mathbf{w}_n^H \mathbf{E}_n^H \mathbf{e}_n = \mathbf{w}_n^H \mathbf{e}_n. \qquad (11)$$

This SINR can be integrated over the range Θ of steering vector angles to find

$$S_\Theta = \int_\Theta \frac{|\mathbf{w}_n^H \mathbf{v}(\theta)|^2}{\mathbf{w}_n^H \mathbf{e}_n} p(\theta) d\theta = \frac{\mathbf{w}_n^H \mathcal{R}_\Theta \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{e}_n}. \qquad (12)$$

[1] An alternative formulation for forming sparse weight vectors is to solve $\mathbf{R}(*, \mathbf{p}_n)\tilde{\mathbf{w}}_n = \mathbf{e}(\mathbf{p}_n)$. For some covariance structures, this sometimes yields better solutions at the expense of higher computational cost. Empirically, this is seldom worth the extra cost.

where

$$\mathcal{R}_{\Theta} = \left[\int_{\Theta} \mathbf{v}(\theta)\mathbf{v}(\theta)^H p(\theta) d\theta \right]. \tag{13}$$

For the specific case of a linear array with steering vectors distributed with spherical symmetry around the array, $\mathcal{R}_{\Theta} = \mathbf{I}$ and

$$S_{\Theta} = \frac{\mathbf{w}_n^H \mathbf{w}_n}{\mathbf{w}_n^H \mathbf{e}_n} = \frac{\tilde{\mathbf{e}}_n^H \mathbf{R}_n^{-2} \tilde{\mathbf{e}}_n}{\tilde{\mathbf{e}}_n^H \mathbf{R}_n^{-1} \tilde{\mathbf{e}}_n}. \tag{14}$$

The first form of this expression is the norm-squared of $\mathbf{w}_n$ over the value of the n-th element. The second form shows that this is upper-bounded by the inverse of the minimum eigenvalue of $\mathbf{R}_n$, which again indicates that the $\mathbf{p}_n$ must be chosen to allow noise eigenvalues in $\mathbf{R}_n$. This also shows that $\mathbf{e}_n$ must lie nearly in the noise subspace for best nulling performance.

2.2. System-level SINR

This section explains the sparse element selection criteria based on overall system performance. While the selection of the set $\mathbf{p}_n$ of auxiliary elements for nulling element n is itself a difficult problem, it is not a sufficient basis for forming *all* such sets to provide the best system SINR when forming a full array nulling weight vector $\mathbf{w} = \sum_{n=1}^{N} \mathbf{w}_n v_n$. The interaction of each sparse weight vector must be considered. The overall SINR averaged over all steering vectors in the region Θ is

$$S_{\Theta} = \int_{\Theta} \frac{|\mathbf{v}^H(\theta)\mathbf{W}^H \mathbf{v}(\theta)|^2}{\mathbf{v}^H(\theta)\mathbf{W}^H \mathbf{R} \mathbf{W} \mathbf{v}(\theta)} p(\theta) d\theta \tag{15}$$

The search for the sparse network element selection sets must ultimately provide the best response with this metric in mind. From our experiments we have found the following guidelines for selecting $\mathbf{p}_n$ [6]

- The size of $\mathbf{p}_n$ should exceed the largest jammer rank expected in the system, so that noise eigenvalues are present in $\mathbf{R}_n$.
- The sparse element selection $\mathbf{p}_n$ for each element should span nearly the same aperture as the full array.
- The relative sparse element spacings of $\mathbf{p}_n$ should differ for each sparse element to reduce vulnerability to specific jammer scenarios.
- The set $\mathbf{p}_n$ of sparse elements chosen for each element should have as small an intersection with all other sparse sets $\mathbf{p}_m$ as possible.

These are utilized in the implementation section below.

3. IMPLEMENTATION

The sparse network approach has been successfully demonstrated on a space-time adaptive array processing (STAP) application for airborne clutter nulling [6]. This application has a distinctive covariance structure which can easily be exploited.

Consider a patch of clutter on the ground at azimuth θ (with $\theta = 0$ denoting broadside of the array). The spatial steering vector to this patch can be written as

$$\mathbf{v}(\theta) = \begin{bmatrix} \exp\left\{j2\pi \frac{x_1}{\lambda} \sin\theta\right\} \\ \vdots \\ \exp\left\{j2\pi \frac{x_N}{\lambda} \sin\theta\right\} \end{bmatrix}, \tag{16}$$

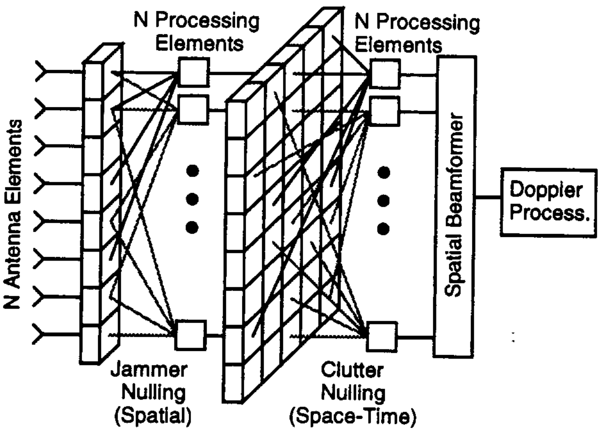

Figure 1. Sparse network implementation of for space-time clutter nulling on an airborne radar platform

where x_i is the position of the i-th element. Due to the platform motion, this clutter patch will have a Doppler frequency $f_d(\theta) = \frac{2v}{\lambda}\sin\theta$ for platform velocity v. The received vectors from M radar pulses can be concatenated into NM-element space-time vectors of the form

$$\tilde{\mathbf{v}}(\theta_k) = \mathbf{d}(\theta) \otimes \mathbf{v}(\theta) \tag{17}$$

where $\mathbf{d}(\theta) = \left[\; e^{j2\pi f_d(\theta)} \;\; \ldots \;\; e^{j2\pi M f_d(\theta)} \; \right]^T$. When all clutter patches are considered, the resulting $NM \times NM$ covariance matrix has been shown to be low rank with the degrees of freedom on the order of the $(N + M)$ [3]. The covariance element corresponding to the p-th element and the m-th pulse with the q-th element and the n-th pulse is

$$r\left((m-1)M + p, (n-1)M + q\right) = \int p(\theta) \exp\left\{ j\frac{2\pi}{\lambda}\left[(x_p - x_q) + 2v(m-n)\right]\sin\theta \right\} d\theta \tag{18}$$

where $p(\theta)$ is the power of the clutter patch at angle θ. For narrow transmit beams, $p(\theta)$ will be focused in azimuth and the above covariance will how very strong symmetry along the line $(x_p - x_q) - 2v(m - n)$ in space-time.

An example SNAP architecture for space-time adaptive processing is shown in Figure 1, which shows a two-stage nulling procedure. The first stage is a sparse network arrangement to null spatial interference such as jamming. The jammer-whitened inputs to the space-time clutter nulling stage are applied to sparse network processors which perform clutter nulling. The sparse element selection for clutter nulling takes place over the two-dimensional element-pulse domain using other sparse element-pulse selections to null each element-pulse pair. The same element-pulse selection scheme may be used on other pulses by treating the resulting sparse network as a two-dimensional temporal filter. After applying this filter, we will produce clutter-free element data on a pulse by pulse basis which can then undergo spatial beamforming followed by Doppler processing.

To prove the usefulness of SNAP for this application, we have simulated its performance with a 32-element uniform linear array using 32 radar pulses and a platform speed and pulse repetition frequency such that the clutter has Doppler ambiguity (i.e., several azimuths have the same Doppler frequency). The angle of the

Table 1. Computation count comparison (in megaflops) for SNAP versus PRI-staggered Post-Doppler

OPERATION:	PRI-Stag.	SNAP
Doppler Proc.	116	
Adapt:	226	25
Apply:	25	67
Doppler Proc.		1
TOTAL	367	93

array is offset 10° from the ground velocity vector. This causes the frontlobe clutter at a given Doppler frequency to have a different azimuth from the backlobe clutter at the same Doppler frequency, resulting in a splitting of the clutter line. We have included jammers at −25° and 15°. As shown in (18), the clutter covariance matrix exhibits a strong banded structure due strictly to the geometry of the airborne radar and its ground velocity, which changes slowly with the flight scenario. Therefore, we may assume a deterministic model for the clutter statistics from which to choose the sparse element-pulse pair selection for element clutter nulling. Deviation from these statistics with real data is compensated by the adaptive weight formation for each element-pulse selection.

We designed an adaptive sparse network processor which used seven pulses centered around the pulse to be nulled. This allowed $7N = 238$ possible degrees of freedom. Sparse network selection sets $\mathbf{p}_n$ employing only 32 degrees of freedom to null each element were chosen using a simulated annealing algorithm with a finite-time cooling schedule [8] to select the sparse selection sets. The sparse element selection criterion provided the best system-level SINR for steering vectors in the interval $|\theta| < 20°$ using an ideal estimate of the clutter covariance matrix. For our simulation, we formed each adaptive sparse element nulling weight vector using only 64 sample vectors (twice the 32 sparse degrees of freedom) with the same distribution as our $NM = 1024$ dimensional covariance matrix. The results of this sparse network application are shown in Figure 2, which shows the SINR loss versus azimuth and Doppler compared to a scenario with no jamming or clutter. The diagonal lines sweeping through the figure are due to front lobe and backlobe clutter, since the array was crabbed 10°. The vertical lines at −25° and 15° are due to jammers at those locations. The jammers affect all Doppler frequencies since they are uncorrelated from pulse to pulse. In the angular sector $|\theta| < 45°$, this SINR performance is nearly identical to the optimal performance. We can compare this algorithm to a represntative space-time architecture called PRI-staggered Post-Doppler [3], which yields similar SINR performance. For a 1024 range gate sample set, SNAP saves a factor of four in overall computations and a factor of ten in adaptive computations, as shown in Table 1. The computational breakdown includes Doppler processing (performed on all data for PRI-staggered Post-Doppler and only on the adapted beamformer output for SNAP), adaptation, and weight application.

At extreme angles ($|\theta| > 45°$), the performance in Figure 2 is noticeably degraded with extraneous nulls appearing between the two clutter nulls. This occurs because these areas have appreciably different *a priori* clutter statistics than the area $|\theta| < 20°$ for which the sparse element selection was chosen. When the sparse element selection was chosen for these sectors, we achieved nearly optimal performance there as well.

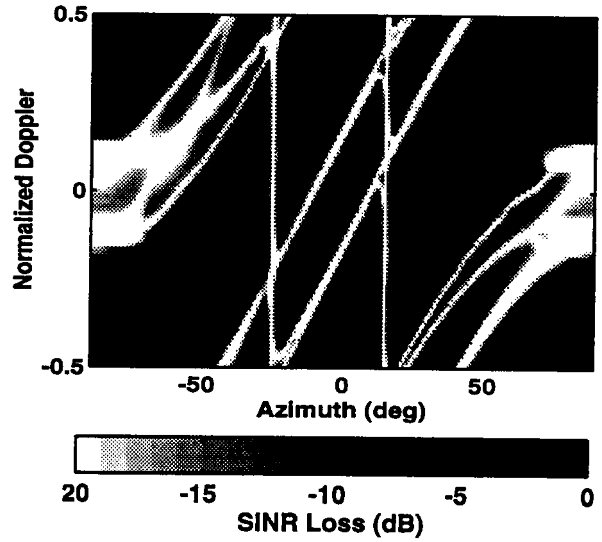

Figure 2. SINR loss for space-time nulling of a 32-element, 32-pulse radar

4. CONCLUSION

In this paper we have demonstrated the Sparse Network Array Processing (SNAP) algorithm which exploits *a priori* covariance structure to yield near optimal performance with significant computational savings. The algorithm nulls the interference on an element-by-element basis using only a fraction of the remaining elements as auxiliary inputs. The architecture is modular, easily scalable, and highly parallel. We have successfully demonstrated the algorithm for space-time adaptive jamming and clutter cancellation for an airborne radar application. While more research is still required to refine the sparse element selection criterion, the algorithm shows great potential for highly structured covariance applications.

REFERENCES

[1] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*. New York: J. Wiley & Sons, 1980.

[2] R. Klemm, "Adaptive clutter suppression for airborne phased array radars," *IEE Proc., Parts F and H*, vol. 130, pp. 125–132, February 1983.

[3] J. Ward, "Space-time adaptive processing for airborne radar clutter suppression," Tech. Rep. TR-1015, M.I.T. Lincoln Laboratory, Lexington, MA, November 1994.

[4] D. Johnson and D. Dudgeon, eds., *Array Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[5] I. Scott and B. Mulgrew, "A sparse approach in partially adaptive linearly constrained arrays," in *IEEE Proc. ICASSP*, vol. IV, (Adelaide, South Australiz), pp. 541–544, April 19-22 1994.

[6] E. J. Baranoski, "Sparse network array processing," *7th IEEE Stat. Sig. Arr. Proc Workshop*, pp. 145–148, June 27-29 1994.

[7] S. L. Marple, Jr., *Digital Spectral Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[8] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. NY: J. Wiley & Sons, 1989.