USC-SIPI REPORT #131

Maximally Sparse Optimization for Array Beamforming and Other Applications

Richard Leahy and Brian Jeffs

Maximally Sparse Optimization for Array Beamforming and Other Applications

Richard Leahy[†]

Brian Jeffs[†‡]

[†]Signal and Image Processing Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

[‡]Hughes Aircraft Company
7000 Village Drive
Buena Park, CA 90621

ABSTRACT

In this paper we introduce algorithms for finding optimally sparse solutions subject to linear and quadratic constraints. The problem of finding the maximally sparse solution, i.e. the vector with fewest non-zero terms, is shown to be closely related to l_p optimization for $0<p<1$. For linearly constrained problems, it is shown that the optimal solution lies at the extreme points of the feasible set. A non linear simplex algorithm is presented for efficiently locating a local minimum on the connected graph of basic feasible solutions. By modifying the algorithm to include a stochastic search, it is shown that a global minimum of the problem may be found. For quadratic constraints, the maximally sparse solution is found by means of a convex transformation. The utlility of these algorithms is demonstrated in the design of arbitrarily shaped, narrowband beamforming arrays. Using the maximaly sparse criteria we are able to peform optimal array thinning and element placement for this problem for an arbitrary set of linear response constraints. The algorithms also have many other applications in signal processing, one such application considered in this paper is the recovery of sparse reflectivity sequences in seismic deconvolution.

EDICS: 4.5 [other suitable categories: 2.3 and 4.4.1]

Submitted to IEEE Transactions on Acoustics Speech and Signal Processing
September 1st, 1988

This work supported in part by the Hughes Aircraft Company Doctoral Fellowship Program.

1. Introduction

We address the problem of finding the maximally sparse solution vector to a system of linear constraints, i.e. a feasible solution with the maximum number of zero valued elements. This problem can be expressed as the nonlinear mathematical program:

$$\min_{x} \; f(x) = \sum_{i=1}^{N} \mathbf{I}(x_i) \quad \textit{such that} \quad |\mathbf{H}\,x - b| \leq \epsilon \tag{1}$$

$$\textit{where} \;\; \mathbf{H} \in R^{M \times N}, \;\; \text{and} \;\; \mathbf{I}(x_i) = \begin{cases} 1 & x_i \neq 0 \\ 0 & x_i = 0 \end{cases}$$

The utility of this class of solutions became apparent to the authors while trying to reconstruct, from the externally measured magnetic fields, 3-d neuromagnetic images of brain neuron currents [1]. With this and other applications discussed in section 5 it was found that the common approaches such as l_2, l_∞, or maximum entropy optimization produced poor results. It is suggested that a large class of signal processing and more general problems exist which would benefit from application of the minimum order criterion. In any problem where extremely "spiked" results are expected, or where the incremental cost of adding a non-zero term to the solution outweighs the cost of increasing an already non-zero term, the minimum order solution is desirable.

The algorithms for minimum order optimization discussed below are based on the related nonlinear program

$$\min_{x} \; g(x) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} \quad \textit{such that} \quad |\mathbf{H}\,x - b| \leq \epsilon, \quad q > 1 \tag{2}$$

Since $[(x)]^q = \|x\|_p$, the l_p quasi-norm of x for $p = \frac{1}{q}$, minimization of $g(x)$ is equivalent to l_p optimization for $0 < p < 1$. We will show that for $q \gg 1$, $g(x)$ approximates $f(x)$ in eqn (1), and provides a more suitable objective for iterative

optimization. We will refer to eqn (2) as an $l_{1/q}$ program.

Previous work related to the topic of this paper includes both mathematical optimization for concave cost functions, and applications where sparse solutions were sought. In [2] the basic behavior of linearly constrained l_p optimization problems for $0<p<1$ is discussed, including examples of how the solution changes in a stepwise fashion as p is varied over this range. The theory of quasi-Banach spaces based on the quasinorm l_p, $0<p<\infty$, has been studied, and is discussed, for example, in [3]. Equation (2) is also related to the linearly constrained concave minimization problem, for which a number of global optimization algorithms have been proposed, based on collapsing polytopes [4] and branch and bound procedures [5,6]. While these methods do achieve global optima, they are probably computationally infeasible for the large dimensions ($N \sim 100$) considered in this paper, due to the use of multiple nested linear programming subproblems and the assumptions necessary to apply them to the general form of eqn (2).

For blind deconvolution of seismic reflectivity data, several authors have discussed the need for optimization norms which increase sparseness or minimize the entropy of the solution. These authors have proposed the use of the "varimax," "parsimonious," and l_p, $p<1$ norms [7,8,9]. These examples are closely related to our problem, however the algorithms used do not in general find the global optimum, and are not directly applicable to the form of eqn (2).

In the area of sparse image reconstruction, linear programming has been applied successfully [10] but without explicitly seeking the maximally sparse image. Also, a technique of "beam subtraction" has been employed to restore sparse astronomical star field images [11]. Design of sparse beamforming arrays, or array "thinning," has also appeared in the literature [12,25] but we know of no published approach for optimally thinned arrays of arbitrary shape.

In the following we describe three algorithms for finding maximally sparse solutions and discuss their properties. They are demonstrated in application to the design of maximally sparse beamforming arrays and the recovery of sparse reflectivity sequences for seismic deconvolution.

2. Minimum Order From l_p Optimization

As an optimization problem, eqn (1) is particularly difficult to solve. We are plagued with numerous local minima, and $f(x)$ is discontinuous and has zero gradient except at the discontinuities. In an effort to overcome these limitations, we propose an approach to the minimum order problem based on generalized linearly constrained l_p optimization. Figure 1 illustrates the unit ball surfaces in R^2 space for the quasi-norm $||x||_{l_p} = [\sum_{i=1}^{N} |x_i|^p]^{1/p}$ for values of p in the range $0 \leq p \leq \infty$.

For $p \geq 1$ we have the conventional l_p norm, which is a convex functional and obeys the triangle inequality. Since the linear constraints in (2) form a convex set it is well known that any local minimum of $||x||_{l_p}$ satisfying the constraints is a global optimum. Many efficient algorithms exist for solving such problems [13,14]. Of particular interest are the cases for values of $p = 1$, 2, and ∞, which form the basis of many widely used optimization procedures. However the resulting solutions for these do not achieve the 'sparse' results of interest in this paper.

For $0 < p < 1$, l_p is only a quasi-norm [3], since the triangle inequality does not hold, and in fact the inequality is reversed for positive x_i. Over R^N, $||x||_{l_p}$ is neither convex nor concave, containing many strong local minima and presenting a difficult optimization problem. Large values of p result in smooth solutions, however, as $p \rightarrow 0$ the solutions tend to become more 'spikey,' or sparse [7].

The reason for this can be seen in Figure 1. As $p \rightarrow 0$, the curves in Figure 1 approach the x_1, x_2 axes, on which the unit ball lies for $f(x)$ in eqn (1). We identify minimum order optimization as a special case of generalized l_p optimization.

Since $g(x)=[\,\|x\|_{l_p}]^p$ for $p=\frac{1}{q}$, we have

$$\lim_{q\to\infty} g(x) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} = \sum_{i=1}^{N} \mathrm{I}(x_i) = f(x)$$

The utility of this observation is that for q finite, $g(x)$ eliminates some of the handicaps of $f(x)$. $g(x)$ is continuous everywhere and differentiable except at the axes. Gradients may be computed for all non-zero terms. This enables use of gradient search techniques at least for finding local minima. In section 3 we present a finite extreme point search algorithm which also benefits from the use of $g(x)$ rather than $f(x)$. For reasonably small values of q, $g(x)$ is computationally stable, and thresholds are not needed to handle inexact zero values. Also, $g(x)$, unlike $f(x)$, can provide some discrimination in cost between solutions of equal order, thus avoiding a stalled search at a point surrounded by adjacent solutions of equal cost.

If however we must allow $q\to\infty$ before eqn (2) leads to a solution of eqn (1), then we cannot benefit from the practical advantages of $g(x)$ mentioned above. Theorem 1 (see Appendix A) provides justification for minimum order optimization based on minimizing $g(x)$ by demonstrating that for a bounded solution set there exists a finite q_1 such that for all $q>q_1$, any solution to eqn (2) is a solution to eqn (1). Eqn (2) therefore defines a class of problems, indexed by q, whose solutions are increasingly sparse as q increases, until $q>q_1$, where an optimally sparse solution is given. For $q\leq 1$ the optimal x changes continuously as a function of q, but for $q>1$, there is a finite number of optimal solutions. A given x_{opt} will remain optimal over a range of q values, and as q increases, we step from one solution to another in a discrete fashion [2]. This behavior is shown in Figure 2 for a problem with four basic feasible solutions, in this case there are two distinct optimal solutions for $0<p\leq 1$.

It should be noted that since solutions to eqn (1) are not necessarily unique, and lacking any justification for accepting one over another, we are satisfied with any algorithm which will select one from the optimal set. Theorem 1 proves that solutions to eqn (2), for $q > q_1$, form a subset of solutions to eqn (1), so we accept any $l_{1/q}$ optimum. In order to improve the computational stability of an algorithm, we wish to use the smallest value of q which reasonably approximates $f(\underline{x})$. The q_1 as computed in Appendix A is a conservative upper bound, and in practice a much smaller value may often be used.

Figure 3 is an illustrative example of the effect of changing q on solving eqn (2) for

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & .2 & 1 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad \underline{\epsilon} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3}$$

Yielding the optimal solutions for various q as shown in Table 1. It is interesting to note that for this example the bound q_1 from Theorem 1 is correctly predicted with $\Omega = 10$, $\epsilon = 1$, and $r = 1$ to be $q_1 = \log(10/1)/\log(2) = 3.32$.

3. Simplex Search Algorithms for l_p Optimization

In the following, we present two algorithms for solving eqn (2) based on a nonlinear simplex search procedure. The $l_{1/q}$ simplex algorithm efficiently locates a local optimum which in practice, appears to provide a good approximation of the global optimum. The stochastic search algorithm converges asymptotically to the global optimum. To facilitate the development, we introduce two specific forms of eqn (2) for $\underline{\epsilon} = 0$ and $\underline{\epsilon} \neq 0$ as follows.

For $\underline{\epsilon} = 0$ the form is:

$$\min_{\underline{x}} \; g(\underline{x}) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} \quad s.t. \; \mathbf{H}\underline{x} - \underline{b} = \underline{0}, \; q > 1 \tag{4}$$

Note that unlike linear programming (LP), no positivity constraint on $\underline{x}$ is

needed, as shown by theorem 2 below. This form may be applied directly to a version of the $l_{1/q}$ simplex algorithm which allows bipolar valued variables.

For $\underline{\epsilon} > 0$ the form of (2) is

$$\min_{\tilde{x}} \; g(\tilde{x}) = \sum_{i=1}^{2N} (\tilde{x}_i)^{\frac{1}{q}} \quad s.t. \;\; \tilde{\mathbf{H}}\tilde{x} - \tilde{b} = 0, \;\; \tilde{x} \geq 0, \;\; q > 1 \tag{5}$$

where

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} & -\mathbf{H} & \mathbf{I} & 0 \\ \mathbf{H} & -\mathbf{H} & 0 & -\mathbf{I} \end{bmatrix} \quad \tilde{x} = \begin{bmatrix} x^+ \\ x^- \\ s^+ \\ s^- \end{bmatrix} \quad \tilde{b} = \begin{bmatrix} b+\epsilon \\ b-\epsilon \end{bmatrix}$$

$$x^+, x^- \in R^N, s^+, s^- \in R^M, \tilde{x} \in R^{N'}, \tilde{b} \in R^{M'}, N' = 2N+2M, M' = 2M$$

s^+ and s^- are respectively slack and surplus variables as commonly employed in linear programming [14]. Note these variables are not included in the computation of $g(\tilde{x})$. This form allows us to confine the search to the nonnegative orthant of the space $R^{N'}$. Theorem 2C below provides the equivalence of equations (5) and (2) for $x = x^+ - x^-$. The introduction of slack and surplus variables, positivity constraints, and use of a form of the $l_{1/q}$ simplex algorithm which allows pivots only to positive valued solutions, are needed to deal with the inequality in eqn (2).

3.1. Fundamental Theorem of $l_{1/q}$ Programming

In solving the $l_{1/q}$ program, we lack the convexity properties which for l_p programming yield a well posed problem, and imply global optimality from local optimality. Over R^N, the cost $g(x)$ is nonlinear and neither convex nor concave. Our problem would appear hopeless, but for the fortunate fact that for $q \geq 1$ we can limit the candidates to a finite set of "basic" solutions, which are the same as those defined for linear programming. This is indicated by the "Fundamental

Theorem of $l_{1/q}$ Programming" which we present here, with proof in Appendix B. We define a basic solution to eqn (4) or (5) as any x or $\tilde{x}$ with respectively at most M or M' nonzero components. Additionally, we say any $\tilde{x}$ of eqn (5) is "properly basic" if it is basic and $(x^+)^T(x^-) = 0$.

Theorem 2A

Given a problem of the form (4) or (5), if a solution exists, then a basic solution exists.

Theorem 2B

If a global optimal solution to eqn (4) exists, it is a basic solution, and is globally optimal to eqn (2) for $\underline{\varepsilon} = \underline{0}$.

Theorem 2C

If a globally optimal solution to (5) exists, then a properly basic globally optimal solution exists, furthermore, this solution implies $x = x^+ - x^-$ is a globally optimal solution to eqn (2) for $\underline{\varepsilon} > \underline{0}$.

The basic solutions of the forms (4) and (5), are isomorphic with the vertices of the convex polyhedron defined by their constraints in R^N and R^{2N+2M} space respectively [14]. Consequently, we can restrict our search to these vertices, since one must be the optimal solution. There are potentially $\mathbf{O}\binom{N}{M}$ of these vertices, making an undirected search of even this finite set impractical for moderately large M and N. These properties motivate us to use a procedure similar to the linear programming simplex algorithm, traversing the vertices while monotonically reducing the cost. Due to the nonlinear nature of the cost, modifications to the standard LP algorithm are required, and globally optimal solutions are not assured.

3.2. The $l_{1/q}$ Simplex Search Algorithm

As in linear programming, we begin by computing any basic feasible solution as a starting point for the simplex search. The two forms presented above describe the constraint as a vector-matrix linear equality in $R^{M\times N}$ or $R^{M'\times N'}$ space respectively. In the following development we will use $\mathbf{H}\underline{x}=\underline{h}$ of form (4), but $\tilde{\mathbf{H}}\tilde{\underline{x}}=\tilde{\underline{h}}$ from (5) can be substituted. Selecting any M columns of $\mathbf{H}$ for which we can compute a left pseudoinverse, we permute the matrix columns and corresponding elements of $\underline{x}$ to place these as the first M columns. Partitioning $\mathbf{H}$ we have

$$[\,\mathbf{A}\,|\,\mathbf{D}\,]=\mathbf{H},\quad \textit{where}\quad \mathbf{A}\in R^{M\times M},\ \mathbf{A}^{\dagger}=\textit{left pseudoinverse of}\ \mathbf{A} \tag{6a}$$

Multiplying by $\mathbf{A}^{\dagger}$ leads directly to the basic solution $\underline{x}_B$

$$[\,\mathbf{I}\,|\,\mathbf{A}^{\dagger}\mathbf{D}\,]\underline{x}=\mathbf{A}^{\dagger}\underline{h}\ ,\qquad \underline{x}_B=[\mathbf{A}^{\dagger}\underline{h},\,0,\cdots 0\,]^T \tag{6b}$$

For form (5), somewhat more care than indicated above is required in selecting columns for $\mathbf{A}$ to insure $\underline{x}_B\geq\underline{0}$. A linear programming phase one procedure may be used with either form to compute a non-negative initial basic solution if we first negate elements of $\underline{h}$, and corresponding rows of $\mathbf{H}$, to force $\underline{h}\geq\underline{0}$, We identify the variables x_i associated with columns of $\mathbf{A}$ as basic variables, and $\mathbf{A}$ as the basis. An adjacent basic solution is one that is formed by moving (pivoting) one variable out of the basis and one non-basic variable into the basis by swapping a column in $\mathbf{A}$ with one in $\mathbf{D}$, adjusting $\underline{x}$ indices, and recomputing $\mathbf{A}^{\dagger}$.

Equation (6) suggests the structure of the "tableau" used in linear programming to facilitate the pivoting computations [14]. The tableau is formed by augmenting the matrix with the right hand side

$$\mathbf{Y}=[\,\mathbf{I}\,|\,\mathbf{A}^{\dagger}\mathbf{D}\,|\,\mathbf{A}^{\dagger}\underline{h}] \tag{7}$$

The reduced cost row of the LP tableau does not appear in $\mathbf{Y}$ due to the nonlinear cost functional. Since the first M columns are always the identity matrix, they need not be stored. During pivoting, all remaining columns of $\mathbf{Y}$ are updated

using the simple pivoting equations, or with a recursive product form of computing the new inverse $\mathbf{A}^{\dagger}$, both described in many texts [14,15]. Index vectors are maintained to keep track of which variables are in or out of the basis. At any iteration, $\mathbf{Y}^i$, the current basic solution can be read directly from the last column. Two tableaus are said to be adjacent if we can move from one to the other with a single pivot operation, or equivalently, if their sets of basic variables differ by one variable only.

For the $l_{1/q}$ simplex algorithm, we may view the set of tableaus as a connected graph with a node for each tableau satisfying (7).

Let $\mathbf{S}$ = the set of all basic feasible solutions (BFS) to (2).

Let $\mathbf{T}$ = the set of all tableaus, $\mathbf{Y}^i$ associated with $\mathbf{S}$.

Define a graph, $G = (\mathbf{T}, \upsilon)$ where υ consists of pairs $(\mathbf{Y}^i, \mathbf{Y}^j) \in \upsilon$ iff tableau $\mathbf{Y}^j$ can be generated from $\mathbf{Y}^i$ by a single pivot operation (or vice versa).

Each element of $\mathbf{T}$ maps onto an element of $\mathbf{S}$. This graph is connected and has properties discussed in depth in [16]. The fundamental theorem implies we may find an optimum by searching the graph G. The algorithm performs this search by generating a sequence, $\{\mathbf{Y}^i \mid i=1,2, \cdots i_{\max}\}$ which traverses the graph along a path of monotonically non-increasing cost, halting when all adjacent solutions are of greater cost. The algorithm is as follows:

1) Find any initial basic feasible solution.

2) Compute the cost, $\sum x_i^{1/q}$, of the bounded feasible solutions from adjacent tableaus.

3) If no adjacent tableau is of equal or lower cost, terminate, optimum found. Otherwise, if any lower cost tableaus exists, select one and pivot to it. Otherwise, perform an anti-cycling procedure [Bland, Kruse] to pivot to an equal cost tableau.

4) Repeat 2) and 3) to termination.

Note that in step 2) the test for feasibility requires a positivity constraint if form (5) is used.

This algorithm's approach is similar to one described in [2], although we have not seen it applied to the minimum order problem. As with linear programming, a practical computer algorithm must deal with accumulated error from pivoting, find an initial solution, include an anti-cycling procedure, and handle systems of large order.

Our experiments have shown that the 1/q simplex algorithm converges in approximately the same number of iterations as the linear algorithm would for a similar sized system. Since the cost at adjacent tableaus can be computed without pivoting the entire tableau, the processing load involved in computing costs at all adjacencies in step 2) above is equivalent to a single pivot operation. Thus computation time is approximately twice that of the LP simplex algorithm overall. With a convergence time comparable to the $\mathbf{O}(5N)$ iterations of the LP simplex, this algorithm is dramatically more efficient than an exhaustive search.

In general, the algorithm does not guarantee convergence to a global optimum. However, we have found that in most cases acceptably sparse solutions were found. This algorithm is much more efficient than the globally optimum stochastic search algorithm described in section 3.3, so if a good, near optimal solution is acceptable, this algorithm is recommended. It is somewhat surprising that it generally performs so well. In the following discussion we attempt to provide insight into why this is so.

A feasible solution to eqn (2) with fewer than M nonzero components is termed a degenerate solution. It follows then, that our original problem, (1), involves a search for the maximally degenerate solution, and for $q > q_1$ we seek the same solution for eqn (2). In standard LP, degeneracy is handled as a nuisance and so-

called 'anti-cycling' procedures, mentioned above, are employed to avoid related problems. Here, degeneracy is precisely what we are looking for.

The problems associated with degeneracy arise from the fact that zero valued basic variables may be interchanged with non-basic variables resulting in a different tableau, but identical corresponding solutions in x. In the non-degenerate case, there is a one-to-one correspondence between each tableau and its corresponding basic solution. Degenerate solutions have a many-to-one relationship. A degenerate solution is overdetermined, with more than one constraint equation active at once [16]. It is important to consider the effect on algorithm performance of having a finite set of tableaus associated with a single basic solution.

Consider a degenerate BFS, $x_j \in \mathbf{S}$, and its representation in the graph G described above. For each such solution we define a degeneracy subgraph, DG^j containing all nodes which map onto x_j and their interconnecting arcs. Consider for example the following system with 5 variables and 3 contraints:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1/2 & 1/6 \\ 0 & 1 & 0 & 1/2 & 1/6 \\ 0 & 0 & 1 & -1/2 & 1/6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \tag{8}$$

The set of basic feasible solutions, their degeneracy degrees and the number of tableaus in the degeneracy subgraph are shown in Table 2 and the associated graph, including the degeneracy subgraphs associated with the degenerate solutions, are shown in Figure 4. Note that the tableaus in DG_3 are not all mutually adjacent. This figure also illustrates the increased number of paths to a degenerate solution. The number of nodes in DG^j increases extremely rapidly as the order of degeneracy increases, and it is likely that many, if not most, of these nodes are not mutually adjacent [16]. This poses a problem for the $l_{1/q}$ simplex algorithm

if such a subgraph is encountered prior to reaching the optimal solution. All nodes in DG^j are of the same cost, yet we must traverse the subgraph to insure access to any lower cost nodes in G which are adjacent to distant nodes of DG^j. One cannot use a single pivot operation to reach all BFS's connected to the degeneracy subgraph associated with x^j. Procedures must be employed to insure that we do not terminate prematurely in DG^j without accessing all external nodes connected to the subgraph, of potentially lower cost, and that we do not 'cycle' endlessly in DG^j. A number of approaches to this problem have been used, including a very simple anti-cycling procedure due to Bland [14,7], perturbation methods [14,17], and algorithms to find the minimum spanning tree for DG^j to guarantee no cycling and access to each external connected point [16]. The stochastic search algorithm described in section 3.3 randomizes the selection of adjacent nodes to pivot to, and thus eliminates the problem of cycling or missing lower cost adjacencies at a degenerate solution.

Degeneracy of the optimal solution however provides advantages which contribute to the success of the algorithm. As the $l_{1/q}$ simplex algorithm traverses the graph from an arbitrary starting point, it can find the global optimum only if a path of monotonic decreasing cost exists through adjacencies. With the many to one mapping of DG^j onto x_j, there are many more paths from an arbitrary point in G to a node mapped to a degenerate solution x_j than would be found for a non-degenerate solution. Thus a degenerate optimum is adjacent to more nodes and it is less likely to be an isolated minimum (see the example of Figure 4).

The algorithm's operational modes differ for forms (4) and (5). With (4), the degeneracy properties described above dominate algorithm performance, but with eqn (5), cost reduction is obtained primarily by pivoting zero cost slack variables into the basis. Few examples of degenerate solution results have been observed for form (5), but results are regularly of lower order than the LP solution.

3.3. The Stochastic Search Algorithm

The algorithm described in this section uses the techniques of simulated annealing [18,19] to arrive at globally optimal solutions to problem (2). The algorithm still limits its search to the graph of basic solutions and traverses the graph across adjacent node arcs. A Markov chain is generated using the Metropolis algorithm [18,19] to randomly select from the adjacent solutions.

Using the graph notation described above, let $N(\mathbf{Y}^i) \subset \mathbf{T}$ denote the set of neighbors, by adjacency, in G to the node $\mathbf{Y}^i$. Let $\mathbf{Y}^i$ be mapped onto the BFS $x_j \in \mathbf{S}$, along with all other nodes contained in the degeneracy subgraph DG_j associated with x_j.

The simplex algorithm described above generates the pair of sequences

$$\{\mathbf{Y}^n \mid n=1,2, \cdots n_{\max}\} \rightarrow \{x^n \mid n=1,2, \cdots n_{\max}\} \tag{9}$$

according to the rule $\mathbf{Y}^{(n+1)} \in N(\mathbf{Y}^n)$ such that $g(x^{n+1}) \leq g(x^n)$, where $\{x^n\}$ may have repeated elements due to degeneracy. Thus the cost is nonincreasing at each iteration. We define local convergence as the first element x_j in the sequence such that all tableaus adjacent to the degeneracy subgraph DG_j of x_j have corresponding solutions of higher cost. Ideally we would like to find the global optimum on the graph G, i.e. $x^{n_{\max}}$ such that $g(x^{n_{\max}}) \leq g(x_i)$ for all $x_i \in \mathbf{S}$. In the following we describe how the method of simulated annealing is used to generate a random search pattern which allows escape from local minima on the graph and converges asymptotically to the global minimum.

Since each element of the sequence $\{\mathbf{Y}^n\}$ belongs to the neighborhood of the previous element, as defined by the graph G, it follows that randomizing the choice of $\mathbf{Y}^{n+1}$ from the neighbors of $\mathbf{Y}^n$ will produce a Markov chain. The key to use of the simulated annealing algorithm is to chose a Gibb's measure, $P_T(\mathbf{Y})$ and randomize the updating rule (choice of tableau) such that for a fixed temperature

parameter, T, the resulting Markov chain is homogeneous [19]. The temperature parameter T is introduced to simulate the annealing process so that as $T \rightarrow 0$, the measure $P_T(\mathbf{Y})$ becomes concentrated at the tableaus corresponding to the global minima of $g(x)$ for all $x \in \mathbf{S}$. The Gibb's measure is chosen to reflect the cost function of our problem:

$$P_T(\mathbf{Y}) = \frac{1}{z(T)} \exp\left[\frac{-g(\mathbf{Y})}{T}\right], \quad \mathbf{Y} \in \mathbf{T} \tag{10}$$

where $z(T)$ is the partition function such that $\sum_{\mathbf{Y} \in \mathbf{T}} P_T(\mathbf{Y}) = 1$ and $g(\mathbf{Y})$ denotes the cost $g(x)$ where x is the BFS associated with $\mathbf{Y}$. For a time homogeneous Markov chain, the Gibb's measure must obey [19]

$$P_T(\mathbf{Y}^i) = \sum_{\mathbf{Y}^j \in \mathbf{T}} \mathbf{P}_T(i,j) P_T(\mathbf{Y}^j) \quad \textit{for all } \mathbf{Y}^i, \mathbf{Y}^j \in \mathbf{T} \tag{11}$$

where $\mathbf{P}_T(i,j)$ denotes the one step transition probability matrix. The choice of the updating rule explicitly determines the matrix $\mathbf{P}_T$. In the following we use the Metropolis algorithm and prove that the resulting matrix $\mathbf{P}_T$ satisfies (11) with $P_T(\mathbf{Y})$ as defined in (10). Each point $\mathbf{Y}^i$ has a neighborhood, $N(\mathbf{Y}^i)$, with cardinality $K_i = |N(\mathbf{Y}^i)|$. Let $\delta(\mathbf{Y}^i)$ denote the degeneracy degree of $\mathbf{Y}^i$ [16], i.e. the number of zero valued elements in the BFS x^i corresponding to tableau $\mathbf{Y}^i$, and $\delta_{\max} = \max_{\mathbf{Y}^i \in \mathbf{T}} [\delta(\mathbf{Y}^i)]$. Then $|K_i| \leq (\delta_{\max}+1)(N-M) = |K|_{\max}$ is the maximum number of adjacent tableaus. Let

$$P_T(i,j) = z(i,j) \min\left[1, \exp\left(-\frac{g(\mathbf{Y}^j) - g(\mathbf{Y}^i)}{T}\right)\right] \tag{12}$$

where

$$z(i,j) = \begin{cases} \dfrac{1}{K_{\max}} & \text{if } \mathbf{Y}^j \in N(\mathbf{Y}^i) \\ 1 - \sum\limits_{j' \neq i} P_T(i,j') & \text{if } i = j \\ 0 & \text{if } \mathbf{Y}^j \notin N(\mathbf{Y}^i),\ i \neq j \end{cases} \tag{13}$$

The Metropolis algorithm [18,19] which produces this transition probability is described below. We first verify that eqn (12) satisfies eqn (11). For (11) to hold, it is sufficient to verify the detailed balance equation [19,20]:

$$P_T(\mathbf{Y}^i)\mathbf{P}(i,j) = P_T(\mathbf{Y}^j)\mathbf{P}_T(j,i) \quad \textit{for all} \quad \mathbf{Y}^i, \mathbf{Y}^j \in \mathbf{T} \tag{14}$$

Case 1: For $j \neq i$, $\mathbf{Y}^j \notin N(\mathbf{Y}^i)$ from eqn (13), $z(i,j)=0$ hence $\mathbf{P}(i,j)=0$. If $\mathbf{Y}^j \notin N(\mathbf{Y}^i)$ then $\mathbf{Y}^i \notin N(\mathbf{Y}^j)$, thus both sides are identically zero.

Case 2: For $j=i$, equality clearly holds.

Case 3: For $\mathbf{Y}^j \in N(\mathbf{Y}^i)$, from eqn (10)

$$\frac{P_T(\mathbf{Y}^i)}{P_T(\mathbf{Y}^j)} = \exp\left(-\frac{g(\mathbf{Y}^i)-g(\mathbf{Y}^j)}{T}\right) \tag{15}$$

From eqn (12)

$$\frac{\mathbf{P}_T(j,i)}{\mathbf{P}_T(i,j)} = \frac{z(j,i)}{z(i,j)} \frac{\min\{1, \exp-[g(\mathbf{Y}^i)-g(\mathbf{Y}^j)]\}}{\min\{1, \exp-[g(\mathbf{Y}^j)-g(\mathbf{Y}^i)]\}} \tag{16}$$

$$= \exp\left(-\frac{g(\mathbf{Y}^i)-g(\mathbf{Y}^j)}{T}\right) = \frac{P_T(\mathbf{Y}^i)}{P_T(\mathbf{Y}^j)}$$

thus eqn (14) and (11) are satisfied.

To generate the corresponding time homogeneous Markov chain, we would use the following algorithm [19]:

1) Select initial basic feasible tableau, $\mathbf{Y}^n$, $n=0$.

2) Let $|N(\mathbf{Y}^n)|=K_n$, then select a tableau $\mathbf{Y}^j \in [N(\mathbf{Y}^n) \bigcup \mathbf{Y}^n]$ according to the probability:

$$Pr(\mathbf{Y}^j) = \begin{cases} 1/K_{\max} & \text{if } \mathbf{Y}^j \in N(\mathbf{Y}^n) \\ 1-K_n/K_{\max} & \text{if } \mathbf{Y}^j = \mathbf{Y}^n \end{cases}$$

3) If $\mathbf{Y}^j = \mathbf{Y}^n$, then $\mathbf{Y}^{n+1} = \mathbf{Y}^n$, $n = n+1$, go to 2), else continue.

4) Compute cost $g(\mathbf{Y}^j)$ for new candidate tableau. If $\Delta g = g(\mathbf{Y}^j) - g(\mathbf{Y}^n) < 0$

Then: $\mathbf{Y}^{n+1} = \mathbf{Y}^j$

Else: generate random variable $r \sim \mathrm{U}[0,1]$

$$\text{If } r \leq \exp\left(-\frac{\Delta g}{T}\right)$$

Then: $\mathbf{Y}^{n+1} = \mathbf{Y}^j$.

Else: $\mathbf{Y}^{n+1} = \mathbf{Y}^n$

5) n=n+1, go to 2).

The convergence to the global optimum is achieved by generating a time inhomogeneous Markov chain by monotonically decreasing the temperature parameter T as the iterations proceed. As $T \to 0$ the Gibb's measure, $P_T(\mathbf{Y})$, will converge to the limit

$$P_0(\mathbf{Y}) = \begin{cases} \dfrac{1}{|\mathbf{T}^*|} & \mathbf{Y} \in \mathbf{T}^* \\ 0 & \mathbf{Y} \notin \mathbf{T}^* \end{cases} \tag{17}$$

where $\mathbf{T}^* = \{\mathbf{Y} \in \mathbf{T}: g(\mathbf{Y}) \leq g(\mathbf{Y}^i) \text{ for } \textit{all } \mathbf{Y}^i \in \mathbf{T}\}$. For global convergence of the simulated annealing algorithm, i.e. the sequence $\{x^n\}$ converges with probability 1 to some element $x^* \in \mathbf{S}^*$, it is necessary that the time inhomogeneous Markov chain is strongly ergodic. The algorithm described above, when modified to reduce the temperature T_n monotonically at each iteration, produces a strongly ergodic Markov chain, provided the annealing schedule is of the form [19]

$$T_n = \frac{\gamma}{\log(n + n_0 + 1)} \qquad n = 0,1,2,\cdots \tag{18}$$

where n_0 is any parameter $1 \leq n_0 \leq \infty$. The parameter $\gamma \geq rL$ is shown in [19] to depend on the radius, r, of the graph and l is a Lipshitz-like constant which is a

measure of the maximum local slope of $g(\mathbf{Y})$ on the graph. Explicit formulas are given for r and L in [19], however, in practice these values are too large for reasonable computation times. In implementing the algorithm the initial temperature was chosen arbitrarily, then reduced according to eqn (18) at each iteration.

Since the simulated annealing algorithm exhibits only asymptotic convergence, and yet the graph G has only a finite number of nodes, this algorithm is of practical importance only if its finite time behavior yields improved solutions over the deterministic simplex search. In section 5 we show simulation results obtained by this algorithm.

4. Convex Transformation Gradient Search

In some applications the constraints may be better expressed as an upper bound on the l_2 norm of the deviation of $\mathbf{H}\underline{x}$ from the desired vector $\underline{h}$, as in the quadratic form:

$$\min_{\underline{x}} g(\underline{x}) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} \quad s.t. \ (\mathbf{H}\underline{x}-\underline{h})^T(\mathbf{H}\underline{x}-\underline{h}) \leq \epsilon, \ x_i \geq 0, \text{ and } q>1 \qquad (19)$$

The convex transformation gradient search algorithm presented here is an efficient algorithm which can handle systems of large dimension. Though global optimality of the result is not assured, as in $l_{1/q}$ simplex algorithm, this produces locally optimal solutions which in practice achieve low order.

If the hyper-volume defined by the constraint $(\mathbf{H}\underline{x}-\underline{h})^T (\mathbf{H}\underline{x}-\underline{h}) \leq \epsilon$ does not contain the origin, then the globally optimal solution must lie on the surface of this volume. The surface contains no isolated extreme points, so we must search a continuous surface, rather than a finite set of points for the optimum. For problems of this form we may not define a finite set of basic solutions, and therefore cannot rely on a simplex search approach.

Attempts at straightforward constrained optimization gradient search techniques are doomed by the extremely strong local minima of the objective function $g(\underline{x})$. The convex transformation approach maps the problem into a space which eliminates this problem and improves the computational and numerical aspects of a gradient search by giving us a convex cost objective functional.

For each point in the original space,

$$\{\underline{x} \mid \underline{x} \in R^N,\ x_i > 0\} \text{ we define } \{\underline{y} \mid y_i = 1/p\ \ln(x_i)\} \tag{20}$$

providing a one-to-one mapping of $\underline{x}$ onto $\underline{y}$. Equation (19) then becomes

$$\inf_{\underline{y}}\ h(\underline{y}) = \sum_{i=1}^{N} e^{y_i} \quad s.t. \quad (\mathbf{H}\, e^{p\underline{y}} - \underline{b})^T\, (\mathbf{H}\, e^{p\underline{y}} - \underline{b}) \leq \epsilon, \text{ and } y_i > -\infty \tag{21}$$

where $e^{\underline{z}}$ denotes point by point exponentiation of each element of a vector $\underline{z}$.

Since e^{y_i} is a strictly convex function, and sums of convex functions are convex, $h(\underline{y})$ is a strictly convex functional over $\underline{y}$. Note that although we have to restrict $x_i \neq 0$ ($y_i > -\infty$), we may allow x_i to be arbitrarily close to zero, and thus consider low order solutions as having the largest possible number of elements within an epsilon neighborhood of zero. This transformation is similar to one used in solving the geometric programming problem by transformation to a convex program [21,22]. In our case however, the original convex set defined by the constraint becomes non-convex.

We have been successful in using exterior point, non-linear constrained optimization algorithms in solving the transformed problem, eqn (21), where these algorithms failed with the original form, eqn (19). By using a least squares solution as a search starting point, a sequential quadratic programming algorithm [23] has been successful in yielding very low order solutions for systems as large as 100 or more variables.

At each iteration, cost function and constraint error evaluations, and their gra-

dient vectors, are computed from the current estimate of x. Exact expressions for the gradients of the cost function and constraints have been derived from eqn (20), and the Hessian matrix is then approximated by assuming a local quadratic form and using finite differences from successive constraint gradients. Algorithm termination is accomplished when an error measure of the Kuhn-Tucker conditions is sufficiently small [23]. Execution times in our experiments compare favorably with the $l_{1/q}$ simplex algorithm for similar sized systems. A penalty method approach [13,14] for constrained optimization has also been used successfully in solving eqn (21).

5. Applications

Two classes of problems have been identified to which the minimum order criterion may be usefully applied. The first class includes system design problems in which the dominant cost is the number of non-zero components required to meet a given set of constraints. An example of this class, which is treated below, is the design of an array beamformer to meet a given spatial response using the minimum number of array elements. Other potential applications include the design of minimum computation FIR filters [24] and problems in operations research. The second type of application lies in the processing of signals. In the following we demonstrate results in seismic deconvolution. Other interesting applications include the reconstruction of star source images in radioastronomy [11] and imaging of current dipole sources as a means of locating neural activity in the human brain from external magnetic field measurements [1].

5.1. Sparse Arbitrary Beamforming Array Design

In this section we consider the problems of array element shading and placement for arbitrarily shaped symmetric 3-D arrays in narrow-band phased beamformer operation. The "fewest elements" optimality criterion allows us to reduce the

beamforming processing load by identifying the unnecessary phones for a given beam; a task which is difficult using other design approaches. Also, if we use a fine grid of potential element locations, then the minimum order solution can be used for optimal element placement analysis.

Although a number of line array thinning approaches have been proposed [25,12], they are not applicable to the arbitrary 3-D array, and do not guarantee optimally sparse arrays even in the 1-D case. The literature contains analysis of a number of unusually shaped arrays, including circular, spherical, cylindrical, and conformal configurations which follow the the shape of the supporting vehicle [26,27,28,29]. With these configurations it is often very difficult to determine efficient element placement; attempting to approximate equal spacing can cluster elements in areas which contribute little to array response, and thinning can become a trial and error proposition. Maximizing the target signal to noise ratio with respect to a known noise field [30], linear programming methods, or using a pattern search algorithm [29,31] can yield useful shadings for these arbitrary arrays, but can give no information on how many elements we need, or where they should be placed. The examples below show how the $l_{1/q}$ optimization algorithms can be applied directly without restriction on the array configuration, and how the maximally sparse optimization approach can improve on other thinning methods.

To set up a system for beamforming design using the $l_{1/q}$ simplex or simulated annealing algorithms, we take M samples of the upper and lower response bounds from a dense enough grid on an enclosing sphere to control sidelobe leakage. Let $\mathbf{s}_i$ be the vector of direction cosines to the point on the sphere where the upper and lower response constraints, b_{u_i} and b_{l_i} are sampled. Let $\mathbf{s}_0$ be the vector direction cosines of the MRA, and $\mathbf{r}_j$ the position vector or the jth array element. Let a_j be the computed shade for the jth element, with $\underline{a} = \underline{a}^+ - \underline{a}^-$ where

$a_j^+, a_j^- \geq 0$ so we may obtain positive or negative shade values while using the positive only vectors $\underline{a}^+$ and $\underline{a}^-$ in the algorithm. We require symmetry about the origin to insure a real response value:

$$\mathbf{r}_j = -\mathbf{r}_{N-j-1}, \text{ and } a_j = a_{N-j-1} \tag{23}$$

where N is the number of array elements, therefore we need only solve for $N/2 + 1$ shades (coefficients) in $\underline{a}$. The response at the constraint points is then given by a cosine transform using the matrix with elements:

$$\mathbf{H}_{ij} = \frac{2}{N}\cos\left[\mathbf{r}_j \cdot (\mathbf{s}_i - \mathbf{s}_0)\omega/c\right], \quad \text{for } i=1,\cdots M \quad j=1\cdots N/2+1 \tag{24}$$

where ω is the band center radian frequency and c is the wave propagation speed. We introduce slack vectors $\underline{z}^+$ and $\underline{z}^-$ of length M, and have the final form of the system:

$$H\underline{x} = \underline{b} \; : \; \begin{bmatrix} \mathbf{H} & -\mathbf{H} & \mathbf{I} & \mathbf{0} \\ \mathbf{H}^\ddagger & -\mathbf{H}^\ddagger & \mathbf{0} & -\mathbf{I}^\ddagger \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \underline{a}^+ \\ \underline{a}^- \\ \underline{z}^+ \\ \underline{z}^- \end{bmatrix} = \begin{bmatrix} \underline{b}_u \\ \underline{b}_l^\ddagger \\ c \end{bmatrix} \tag{25}$$

where ‡ indicates rows may have been negated to force $\underline{b}_l^\ddagger$ to be non-negative. The bottom row of H is added to constrain the sum of shade absolute values. Our simulations show c can be adjusted to improve array gain relative to a noise field and beamformer stability. From eqn (25) we first use a phase one algorithm to find any basic solution and then optimize using the simplex or simulated annealing algorithms. With a low order solution $\underline{a}$ computed, the final complex element weight for the beamformer is

$$\gamma_j = a_j \mathbf{e}^{-\mathbf{j}\frac{\omega}{c}(\mathbf{r}_i \cdot \mathbf{s}_0)} \tag{26}$$

For all examples given below, q=15 was used by the optimization algorithms.

Consider the 60 element transparent concentric ring array of Figure 5. We wish to form beams, steered horizontally, in the plane containing the array. This is similar to the configuration used by some "dipping" sonar systems which suspend a cylindrical ring array in the water from a helicopter and form horizontal search beams. Figure 6 shows the beam response for the full array using unity magnitude shading, with complex phase shifting at each element equal to the conjugate of the elemental propagation phase delay for a plane wave arriving from the maximum response angle (MRA) of zero degrees. A sinusoidal signal at 1 kHz is assumed, which gives an average element to element spacing of just over $\lambda/4$. We require the element positions to be symetric about the origin.

For the thinned array design, we use the same element phasing as in Figure 6, but let the algorithm adjust the real amplitude shading. The mainlobe width is constrained to be the same as Figure 6, with sidelobes no larger than the first sidelobe. Allowing some of the secondary sidelobes to come up to the level of the first allows some degree of freedom which is exploited by the algorithm to thin the array. Figure 7a shows the remaining elements of the array after thinning by the $l_{1/q}$ simplex search algorithm, and Figure 7b shows the corresponding response pattern. Only 16 of the original elements are needed to maintain the original mainlobe shape and maximum sidelobe level. This result agrees with earlier observations that the outer elements of a ring array are the primary contributers to beam response. Note that the algorithm simultaneously selects the elements and computes the optimal shade weighting.

In [25], Jarske et. al., propose a simple thinning procedure for narrow beam arrays. In their example 4.1, a symetric line array is designed with length constrained to be $\leq 50\lambda$, and a mainlobe width constraint of +/-3.6 degrees (wavenumber = $.08\pi/\lambda$). The thinning procedure used in [25] requires the elements to be placed at multiples of $\lambda/2$ from the array center. Figure 8a shows

the final element positions for the best solution in [25], which produced a maximum sidelobe level of .217 (-13.27 dB), using 25 elements. Figure 8b shows the positions of the 26 elements, and Figure 9 shows the corresponding response pattern, for the $l_{1/q}$ simplex search solution to the same problem, but with the mainlobe further constrained to +/-2.06 degrees. The initial array used in the search was 251 elements long, with .2λ spacing, for a total length of 50λ. Using the stochastic search algorithm and truncating the sequence prior to reaching a temperature of zero, this solution was improved to that shown in Figures 8c and 10. Note that here only 24 elements were needed, the aperture is slightly smaller, and the mainlobe narrower than example 4.1 of [25]. The difference between the $l_{1/q}$ and the stochastic search solutions is an example of termination at a local optimum which is overcome by the simulated annealing randomization of the search. Figure 8d shows the final element positions, and Figure 11 the corresponding beam response for the $l_{1/q}$ search with all constraints (including mainlobe width) identical to the example in [25]. Only 16 elements were needed and the aperture was reduced further.

5.2. Seismic Deconvolution

The goal of seismic deconvolution is to estimate the acoustical properties of a layered medium by recovering a reflectivity sequence, representing the impedance mismatches between the layers, from a set of measurements of an incident wavelet relected from each of the boundaries [32]. Several methods have been proposed for 'blind deconvolution', in which the source wavelet is unknown, based on the optimization of norms which favor sparse reflectivity sequences. Examples of such norms are the varimax and parsimonius functionals [7,8,9]. While these methods have met with some success, the cost functions are nonconvex, and since the methods are formulated as unconstrained optimization problems [8], the search is carried out over a continuous parameter space, converging to a local optimum.

Methods based on the above cost criteria, do not adapt well to the case in which the source wavelet is known and the problem can be formulated in terms of a set of linear constraints, since local minima are not confined to the vertices of the convex polytope. An alternative approach is taken by Mendel [32] based on a statistical model for the signal and reflectivity sequence. A Bernoulli-Gaussian model describes the reflectivity sequence, in which the parameter, λ, of the Bernoulli process determines the degree of sparseness of the layers and the variance, σ^2, of the Gaussian, the average energy reflected at each boundary.

In the following we apply the methods developed above to the deconvolution problem in which the source wavelet is known and compare the results with those obtained using the optimal deconvolution methods in [32].

The linear system was set up to include slack variables in a similar manner to that described in section 5.1 to allow for noise in the data. The formulation differs slightly since rather than constraining each component of the error vector, $\underline{\epsilon}=(\underline{x}-\mathbf{H}\underline{h})$, an upper bound was placed on the sum of the magnitude of the components of $\underline{\epsilon}$, i.e. $\sum_{i=1}^{M}|\epsilon_i|=\sum_{i=1}^{M}|(z_i^+-z_i^-)|\leq c$. The effect of this modification is to allow larger errors at a few sample points provided the total error is not too large. A suitable choice of c could be found from an estimate of the variance of the noise in the signal. The system which imposes the above the constraints is as follows:

$$H\underline{x}=\underline{h}\ :\ \begin{bmatrix}\mathbf{H} & -\mathbf{H} & \mathbf{I} & \mathbf{0} & \underline{0}\\ \mathbf{H}^\dagger & -\mathbf{H}^\dagger & \mathbf{0} & -\mathbf{I}^\dagger & \underline{0}\\ \underline{0} & \underline{0} & \underline{1} & \underline{1} & 1\end{bmatrix}\begin{bmatrix}\underline{a}^+\\ \underline{a}^-\\ \underline{z}^+\\ \underline{z}^-\\ s\end{bmatrix}=\begin{bmatrix}\underline{h}\\ \underline{h}\\ c\end{bmatrix} \tag{27}$$

where $\underline{h}$ denotes the reflectivity data and $\underline{z}=\underline{z}^+-\underline{z}^-$ the unknown reflectivity sequence.

A seismic signal was simulated by convolving the reflectivity sequence in Fig. 12b, generated according to a Bernoulli Gaussian model(λ=1/16, σ^2=1), with the causal 4th order ARMA wavelet in Fig. 12a. The resulting signal, corrupted with i.i.d. Gaussian noise at an SNR of 10 dB, is shown in Fig. 13a. A second set of data was generated to incorporate convolutional backscatter [32]. In this case an i.i.d Gaussian sequence, of variance σ_b^2=0.01, was added to the reflectivity sequence prior to convolution with the wavelet; again i.i.d Gaussian noise was added to the resulting data at a SNR of 10dB. The data is shown in Fig. 13b.

The results of deconvolution are shown for Mendel's optimal seismic deconvolution method (OSD) (Fig. 14) and the nonlinear simplex algorithm (NSA)(Fig. 15) for the signal plus noise case (Figs. 14a and 15a) and the signal plus backscatter plus noise case (Fig. 14b and 15b). The square boxes in the graphs show the locations of the detected events, the solid lines the locations of the actual events. In the backscatter case, we are interested only in the larger events, the smaller 'events' being due to backscatter from small scatterers and not of primary interest in this problem. When applying the OSD method we assumed only knowledge of the wavelet parameters; all variances and the Bernoulli parameter were estimated from the data. Event detection was performed using the 'single most likely replacement' detector [32].

It is interesting to note that there is very little difference between the results obtained from the two approaches, even though the minimum variance deconvolution was based on the exact statistical model by which the data was generated, while the new method uses only the $1/p$ cost function. In the case for data plus noise the OSD method produces a spurious event at about the 80th sample point (Fig. 14a) which is not present in the NSA solution. In the cae with backscatter, the OSD performs slightly better and detects only 'true' events while the NSA method also detects several of the larger 'backscatter events'. In both cases how-

ever, the number of events detected may be altered by modifying either the level of the threshold in the event detection in OSD [32] or the upper bound, c, on the l_1 norm of the residual error. In either case, an estimate for these values could be obtained from more detailed knowledge of the varainces of the backscatter and additive noise terms.

6. Conclusions

Algorithms have been presented for determining minimum order solutions to problems with linear or quadratic constraints. These algorithms are based on minimization of the l_p norm, $0<p<1$, which for a sufficiently small value of p is equivalent to the minimum order criterion, as shown in Theorem 1. In the case with linear constraints it was shown in Theorem 2, that the solution lies at a vertex of the convex polytope formed by the linear constraints and can therefore be found using a simplex algorithm with either deterministic or random updating resulting respectively in localy and globaly optimal solutions. In the case with quadratic constraints a convex transformation was described. The algorithms were demonstrated to achieve good, sparse solutions in application to beamformer design and seismic deconvolution.

7. ACKNOWLEDGEMENTS

The authors would like to thank Mr. Zhenyu Wu for his suggestions and assistance in obtaining the seismic deconvolution results and Dr. Jerry Mendel for the use of his optimal seismic deconvolution subroutine package. Also, Mr. Thomas Hebert provided helpful suggestions in the development of the optimization algorithms.

8. References

[1] B. Jeffs, R. Leahy and M. Singh, *'An evaluation of methods for neuromagnetic image reconstruction'*, IEEE Trans. Biomed. Eng., vol BME-34, pp 713-723,

1987.

[2] I. Barrodale and F. Roberts, *'Application of mathematical programming to l_p approximation'*, in 'Nonlinear Programming', J. Rosen et al, Eds., Academic Press, 1970.

[3] A. Pietsch, *'Approximation spaces'*, J. Approx. Theory, Vol. 32, pp 115-134, 1981.

[4] J. Falk and K. Hoffman, *'Concave minimization via collapsing polytopes'*, Oper. Res., Vol. 34, pp 919-929, 1986.

[5] P. Zwart, *'Global maximization of a convex function with linear inequality constraints'*, Oper. Res., Vol. 22, pp 602-609, 1974.

[6] N. Thoai and H. Tuy, *'Convergent algorithms for minimizing a concave function'*, Math. Oper. Res., Vol. 5, pp 556-566, 1980.

[7] W. Gray, *'Variable norm deconvolution'*, Ph. D. Thesis, Stanford University, 1979.

[8] R. Wiggins, *'Entropy guided deconvolution'*, Geophys., Vol. 50, pp 2720-2726, 1985.

[9] D. Donoho, *'On minimum entropy deconvolution'*, in 'Applied Time Series Analysis, II', Academic Press, 1981.

[10] R. Mammone, *'Image restoration using linear programming'*, in 'Image Recovery: Theory and Applications', Ed H. Stark, Academic Press, 1987.

[11] U. Schwarz, *'Mathematical statistical description of the iterative beam removing technique, [method CLEAN]'*, Astr. Astrophy., Vol. 65, pp 345-356, 1978.

[12] A. Ishimaru and Y-S Chen, *'Thining and broadbanding antenna arrays by unequal spacings'*, IEEE Trans. Anten. Prop., vol. AP-13, pp 34-42, 1965.

[13] L. Scales, *'Introduction to non-linear optimization'*, Springer-Verlag, New York, 1985.

[14] D. Luenberger, *'Linear and nonlinear programming'*, 2nd Ed., Addison Wesley, 1984.

[15] G. Strang, *'Linear algebra and its applications'*, 2nd Ed., Academic Poress, 1980.

[16] H. Kruse, *'Degeneracy graphs and the neighbourhood problem'*, Lecture notes in Econ. Math. Systems, Vol. 260, Springer-Verlag, 1986.

[17] A. Schrijver, *'Theory of linear and integer programming'*, J. Wiley, 1986.

[18] S. Geman and D. Geman, *'Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images'*, IEEE Trans. Patt. Anal. Mach. Int., Vol. PAMI-6, pp 721-741, 1984.

[19] D. Mitra, F. Romeo and A. Sangiovanni-Vincentelli, *'Convergence and finite-time behaviour of simulated annealing'*, Adv. Appl. Prob., Vol. 18, pp 747-771, 1986.

[20] J. Marroquin, *'Probabilistic solutions of inverse problems'*, Ph.D. Thesis, MIT, 1985.

[21] R. Duffin, *'Linearizing geometric programs'*, SIAM Review, Vol. 12, pp 211-227.

[22] R. Duffin, E Peterson and C. Zener, *'Geometric programming'*, John Wiley, New York, 1967.

[23] K. Schittkowksi, *'NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems'*, Anals Oper. Research, Vol. 5, pp 485-500, 1980.

[24] R. Leahy, B. Jeffs and Z. Wu, *'A DSP algorithm for minimum order solutions'*, Proc. 21st Asilomar Conf. Signals, Syst. Comp., Nov. 1987.

[25] P. Jarske, T. Saramaki, S. Mitra, U Neuvo, *'On Properties and Design of Nonuniformly Spaced Linear Arrays'*, IEEE Trans. Acoust., Speech, Signal

Processing, vol ASSP-36, pp 372-380, 1988.

[26] T. Frank, J. Kesner and H. Gruen, *'Conformal array beam patterns and directivity indices'*, J. Acoust. Soc. Am., V0l. 63, pp 841-847, 1978.

[27] R. Streit, *'Optimization of discrete arrays of arbitrary geometry'*, J. Acoust., Soc. AM., V0l. 69, pp 199-212, 1981.

[28] E. Sullivan, *'Side-lobe behaviour of conformal arrays'*, J. Acoust. Soc. Am., Vol 71, pp 402-404, 1982.

[29] S. Prasad and R. Charan, *'On the constrained synthesis of array patterns with applications to circular and arc arrays'*, IEEE Trans. Anten. Prop., Vol AP-32, pp 725-730, 1984.

[30] L. Griffiths and C. Jim, *'An alternative approach to linearly constrained beamforming', IEEE Trans. Anten. Prop., Vol. AP-30, pp 27-34, 1982.*

[31] R. Hooke and T. Jeeves, *'Direct search solution of numerical and statistical problems'*, J. ACM., Vol. 8, pp 212-229, 1961.

[32] J. Mendel, *'Optimal seismic deconvolution'*, Acad. Press, 1983.

9. Appendix A

Theorem 1: Let S denote the set of all basic feasible solutions to $\mathbf{H}\underline{x} = \underline{h}$ *s.t.* $\mathbf{H} \in \mathbf{R}^{M \times N}$. If the solutions in S are bounded, then $\Omega = \max_{\underline{x} \in \mathbf{S}} [l_\infty(\underline{x})]$ is finite. Let $\epsilon = \min_{x_i \in \mathbf{S}, 1 \le j \le N} \left\{ |x_{ij}| \text{ s.t. } x_{ij} \neq 0 \right\}$ i.e. ϵ is the smallest non-zero magnitude of any element of any vectors in S.

Given $\epsilon > 0$ and $\Omega < \infty$, if V is the set of all globally optimal solutions to

$$\min_{\underline{x}} \; f(\underline{x}) = \sum_{i=1}^{N} \mathbf{I}(x_i) \quad \textit{such that} \quad \mathbf{H}\,\underline{x} = \underline{h} \tag{27}$$

with $r = f(\underline{x})$ for any $\underline{x} \in V$ (i.e. r is the optimal solution order), and U is the

set of all globally optimal solutions to

$$\min_{x} \; g(x) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} \quad \textit{such that} \quad \mathbf{H}\,x = h, \;\; q > 1 \tag{28}$$

then $U \subset V$ if $q \geq q_1$ where

$$q_1 = \frac{\log\left(\frac{\Omega}{\epsilon}\right)}{\log\left(\frac{r+1}{r}\right)} \tag{29}$$

Proof:

1) By Theorem 2A, and since any non-basic solution to eqn (27) is of higher order than a basic solution, $V \subset S$. Also, by Theorem 2B, $U \subset S$.

2) Since $|x_j| \leq \Omega$, $1 \leq j \leq N$ for all $x \in S$, $g(x) \leq r\Omega^{1/q}$ for all $x \in V$

3) Since the order of any basic solution, x', not in V is $\geq r+1$, and $|x_j'| \geq \epsilon$, $1 \leq j \leq N$, for $x_j' \neq 0$, $g(x') \geq (r+1)\epsilon^{1/q}$ for all $x' \in S \bigcap \overline{V}$.

4) If for some q, $r\Omega^{1/q} < (r+1)\epsilon^{1/q}$ then by 2) and 3), $g(x) < g(x')$ for all $x \in V$ and all $x' \in S \bigcap \overline{V}$, then by 1), solutions minimizing $g(x)$ must be contained in V, i.e. $U \subset V$.

5) Solving for q in 4):

$$\frac{r+1}{r}\left(\frac{\epsilon}{\Omega}\right)^{1/q} > 1$$

$$\log\left(\frac{r+1}{r}\right) + \frac{1}{q}\log\left(\frac{\epsilon}{\Omega}\right) > 0$$

$$q > \frac{\log\left(\frac{\Omega}{\epsilon}\right)}{\log\left(\frac{r+1}{r}\right)} = q_1$$

and since $0<r,\epsilon,\Omega<\infty$, q_1 is finite. **Q.E.D.**

Note that this is a sufficient (but not necessary) condition which gives a conservative upper bound on q and that in practice a much smaller value may be used. Even though the equations may contain unbounded or very large finite solutions, the application itself may suggest a more realistic value for Ω. This smaller Ω will yield a smaller q which will minimize the objective, $g(\underline{x})$, for the lowest order realistic solution, but will return a higher cost for lower order, yet unrealistic solutions. For example, in the beamforming problem, we know that for a stable shading design with a normalized MRA response of 1.0, any single shade cannot be significantly larger than N. Mathematically feasible, yet practically unrealistic solutions of low order can be rejected by using the "common sense" value for $\Omega \sim N$. Likewise, a realistic value for ϵ can be inferred from the measurement and quantization noise in our problem.

10. Appendix B

Theorem 2A, Proof:

The existence of a solution, or a basic solution, is dependent only on the constraint equation $\mathbf{H}\underline{x} = \underline{h}$, not on the cost functional. Therefore, this portion of the theorem is equivalent to the linear programming case, for which a proof is available in many texts [Luenberger].

Theorem 2B, Proof:

1) From eqn (4) above, we have the form

$$\min_{\underline{x}} \; g(\underline{x}) = \sum_{i=1}^{N} |x_i|^{\frac{1}{q}} \quad \textit{such that} \quad \mathbf{H}\,\underline{x} = \underline{h}, \;\; q > 1 \tag{31}$$

we construct the system

$$\min_{\underline{y}} \; h(\underline{y}) = \sum_{i=1}^{2N} (y_i)^{\frac{1}{q}} \quad \textit{such that} \quad \mathbf{C}\,\underline{y} = \underline{h}, \;\; \underline{y} \geq \underline{0}, \;\; q > 1 \tag{32}$$

where

$$\underline{y} = \begin{bmatrix} \underline{x}^+ \\ \underline{x}^- \end{bmatrix}, \quad \mathbf{C} = \left[\mathbf{H} \mid -\mathbf{H}\right], \quad \underline{x} = (\underline{x}^+ - \underline{x}^-)$$

2) By construction, any feasible $\underline{y}$ for (32) implies an $\underline{x}$ feasible for (31). Similarly, $\underline{y}$ basic for (32) implies an $\underline{x}$ basic for (31).

3) Over $\underline{y} \geq \underline{0}$, $\sum_{i=1}^{2N} (y_i)^{1/q}$ is strictly concave, and $\mathbf{C}\underline{y}=\underline{h}$ defines a convex set. Any strictly concave functional achieves its global minimum at an extreme point of a convex constraint set. Since a solution is an extreme point iff it is basic [Luenberger], then eqn (32) is minimized at some basic solution, $\underline{y}_{opt}$ with cost $h(\underline{y}_{opt})$.

4) Given $\underline{y}_{opt} = [\underline{x}^+_{opt}, \underline{x}^-_{opt}]^T$ let $\hat{\underline{x}} = \underline{x}^+_{opt} - \underline{x}^-_{opt}$. We prove by contradiction $\underline{y}_{opt}$ is properly basic. Assume that for a pair of terms in $\underline{y}_{opt}$, $(x^+_{i_{opt}}) \cdot (x^-_{i_{opt}}) \neq 0$, and thus $\underline{y}_{opt}$ is not properly basic. We may form a new vector, $\underline{y}'$ by replacing only these terms with:

$$x_i^{+\prime} = x^+_{i_{opt}} - min(x^+_{i_{opt}}, x^-_{i_{opt}}) \text{ and } x_i^{-\prime} = x^-_{i_{opt}} - min(x^+_{i_{opt}}, x^-_{i_{opt}})$$

By inspection $\underline{y}'$ is a basic feasible solution to (32) and $h(\underline{y}') < h(\underline{y}_{opt})$ since $\underline{y}'$ differs from $\underline{y}_{opt}$ in only two terms, which are both smaller than the corresponding terms in $\underline{y}_{opt}$. This contradicts the optimality of $\underline{y}_{opt}$, so we conclude $(\underline{x}^+_{i_{opt}})^T (\underline{x}^-_{i_{opt}}) = 0$.

5) $g(\hat{\underline{x}}) = h(\underline{y}_{opt})$ for any optimum solution to (32) since due to 4)

$$h(\underline{y}_{opt}) = \sum_{i=1}^{2N} (y_{i_{opt}})^{1/q} = \sum_{i=1}^{N} (x^+_{i_{opt}})^{1/q} + \sum_{i=1}^{N} (x^-_{i_{opt}})^{1/q} = \sum_{i=1}^{N} |\hat{x}_i|^{1/q} = g(\hat{\underline{x}})$$

6) We prove by contradiction that $\hat{x}$ is optimum for (31). Assume $\hat{\underline{x}}$ is not optimum for eqn (31), but some $\ddot{\underline{x}}$ is, with $g(\ddot{\underline{x}}) < g(\underline{x}_y)$. We could then construct a feasible solution $\ddot{\underline{y}} = (\ddot{\underline{x}}^+, \ddot{\underline{x}}^-)^T$, with $\ddot{\underline{x}}^+$ containing the positive terms of $\ddot{\underline{x}}$ and $\ddot{\underline{x}}^-$ the negative terms. $(\ddot{\underline{x}}^+)^T (\ddot{\underline{x}}^-) = 0$, so as in 5)

$g(\ddot{x}) = h(\ddot{y}) < h(y_{opt}) = g(\hat{x})$ which contradicts the assumption y_{opt} is optimal for (32). Therefore, $\hat{x}$ is optimal for (31).

7) Thus, given y_{opt} optimum for (32), by 3) it must be basic, 2) implies an $\hat{x}$ which is also basic for (31), which by 6) is also optimum. **Q.E.D.**

Theorem 2C, Proof:

1) $\tilde{\mathbf{H}}\tilde{x} - \tilde{b} = 0$, $\tilde{x} \geq 0$ defines a convex solution set over which $g(\tilde{x})$ is concave. If an optimal solution $\tilde{x}^0$ exists, then, since $g(\tilde{x})$ is (not strictly) concave, there exists a solution $\tilde{x}_{opt}$ which is an extreme point, therefore basic, with $g(\tilde{x}^0) = g(\tilde{x}_{opt})$.

2) We may use an argument similar to step 4) of Theorem 2B above to prove $(x_{opt}^+)^T(x_{opt}^-) = 0$ and thus $\tilde{x}$ is properly basic.

3) Since s^+ and s^- are not included in the cost computation $g(\tilde{x})$, as in 5) and 6) above $g(\tilde{x}_{opt}) = g(x)$, $x = x_{opt}^+ - x_{opt}^-$, and hence if $\tilde{x}_{opt}$ is an optimal solution to (5), x is an optimal solution to (2). **Q.E.D.**

Optimal Solutions				
q values:	Solution type:	x_1:	x_2:	x_3:
0	min-max	.91	.91	.91
.5	min energy	.98	.20	.98
$1 \leq q \leq 3.32$	linear program	1.00	0.00	1.00
$q > 3.32$	min order (max sparse)	0.00	10.00	0.00

Table 1: Optimal solutions to eqn (2) for the system of eqn (3) for various q.

BASIC FEASIBLE SOLUTION:						Order of Degeneracy	Tableaus in Degeneracy Subgraph
	x_1	x_2	x_3	x_4	x_5		
x_1	1	1	1	0	0	0	(1)
x_2	2	2	0	-2	0	0	(2)
x_3	0	0	0	0	6	2	(3),(5),(6),(7),(9)
x_4	0	0	2	2	0	1	(4),(8)

Table 2: Comparison of the degeneracy of the feasible basic solutions to eqn (8)

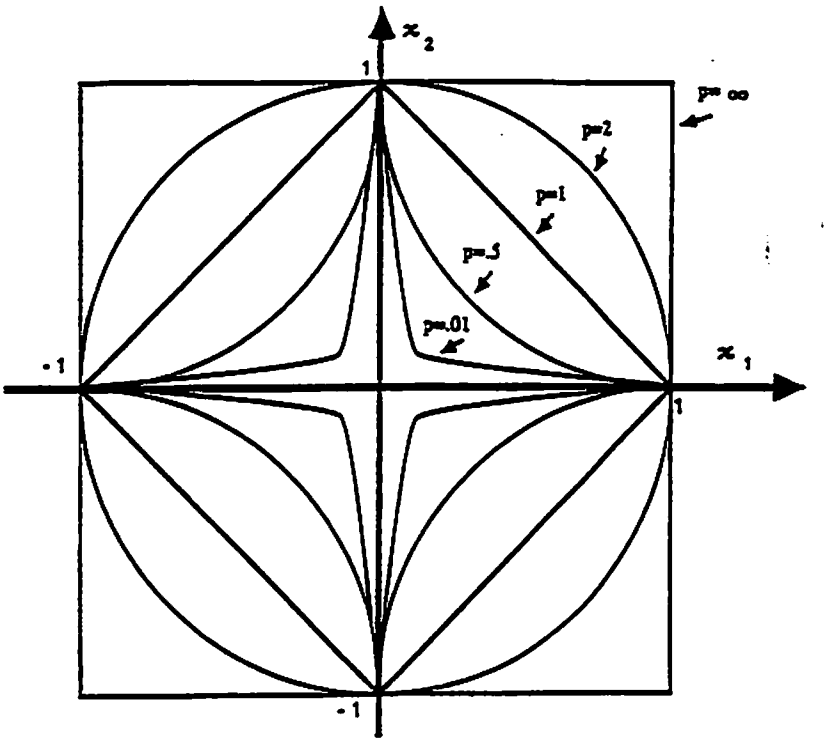


Figure 1. Unit balls of the l_p norm for various p. Note as p approaches 0 the unit ball approaches the axes.

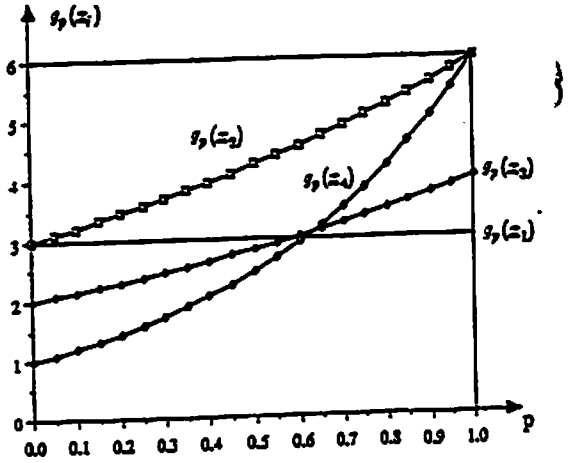


Figure 2. Illustration of the cost, $g_p(x_i)$ associated with each of the four basic feasible solutions to the problem of eqn (4) for p values of $0<p\leq 1$. Note for $0<p\leq .613$, x_4 is of lowest cost, while for $.613<p\leq 1$, x_1 is of lowest cost.

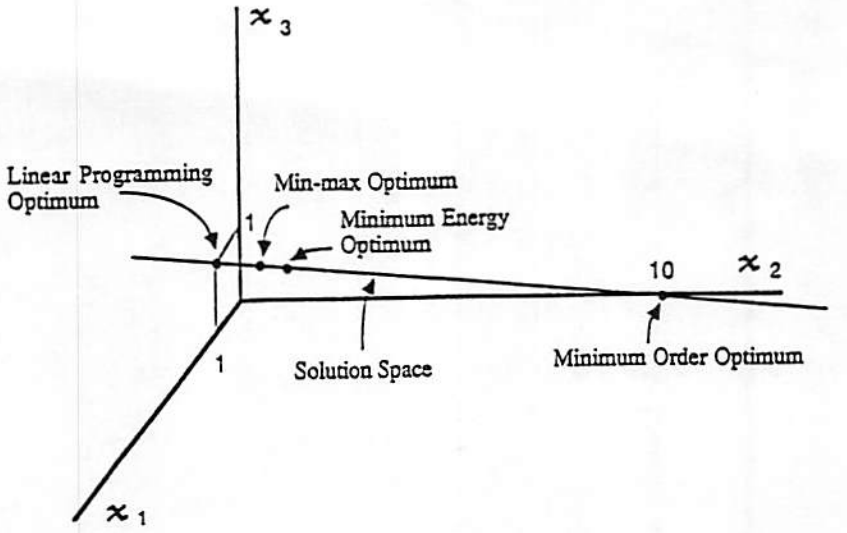


Figure 3. An example of l_q optimization for various values of q. Solutions to equation (3) show minimum order results for q>3.32

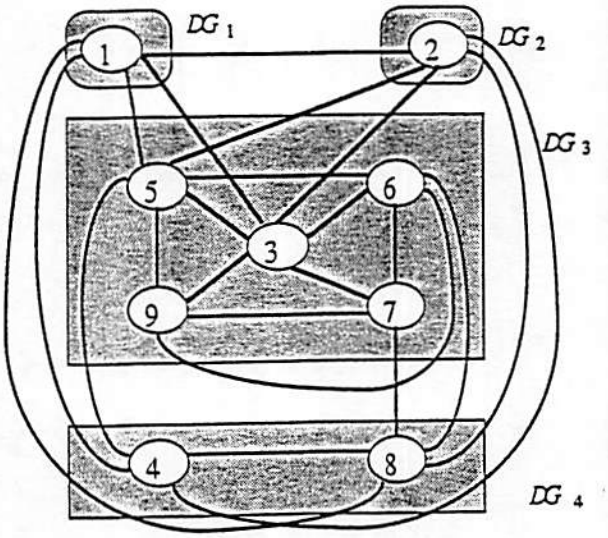


Figure 4: Graph of the simplex tableaus and their connections for the example in eqn (8). Each shaded area, DG_i , represents the degeneracy subgraph associated with basic feasible solution Xi. Graph node numbers indicate the corresponding unique tableaus.

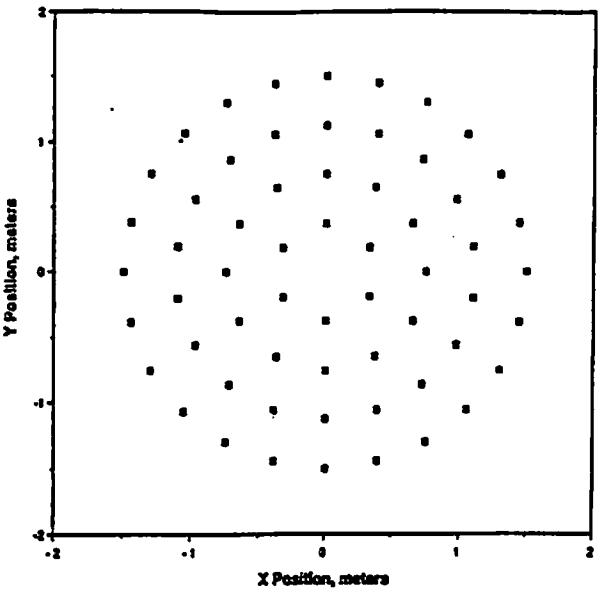


Figure 5. Original 60 elements concentric ring array. Elements are omnidirectional and beams are formed in the plane of the array. Design is for 1 kHz acoustic operation in seawater.

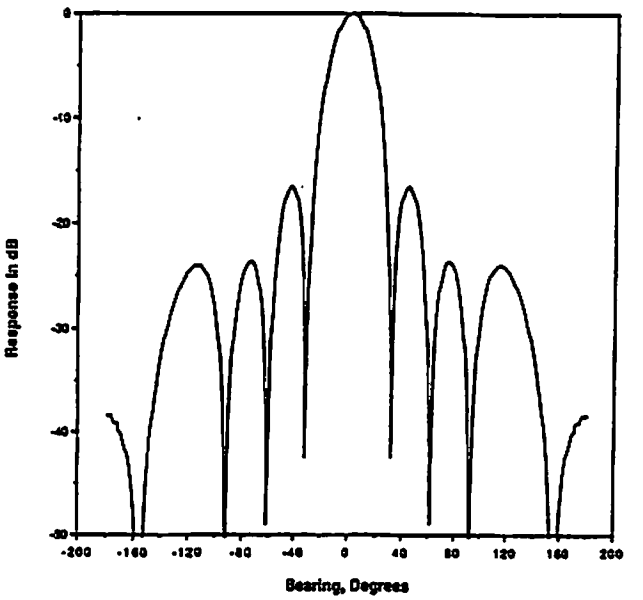
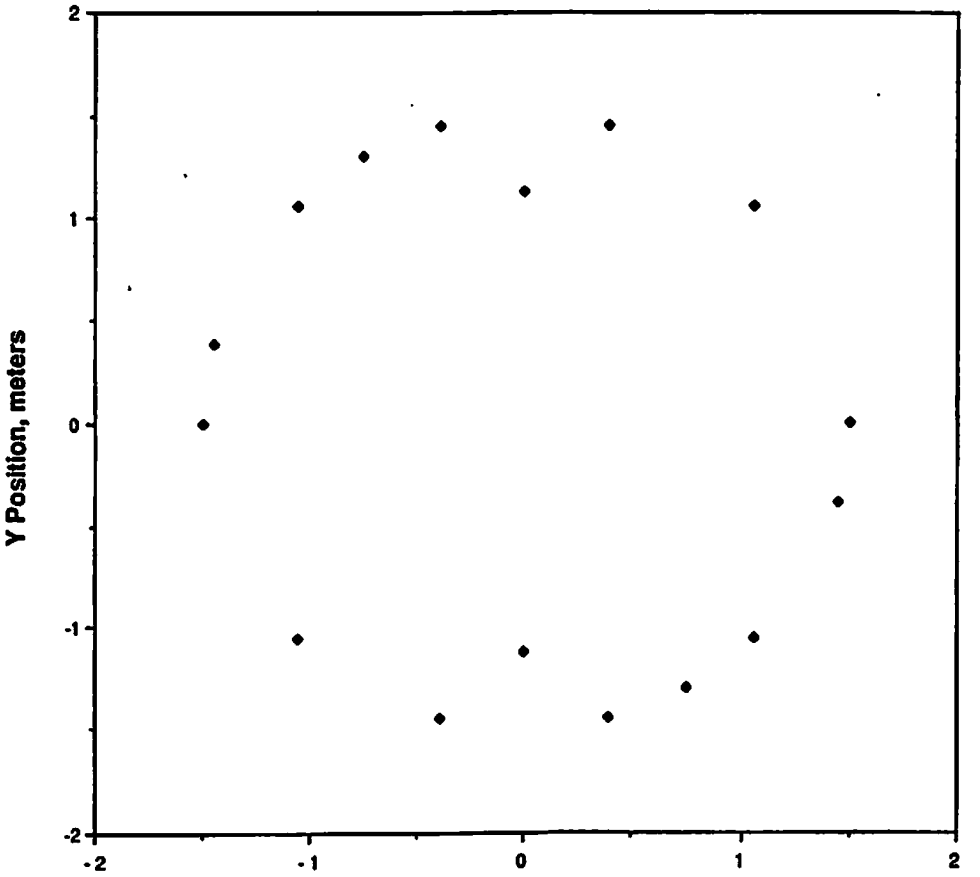
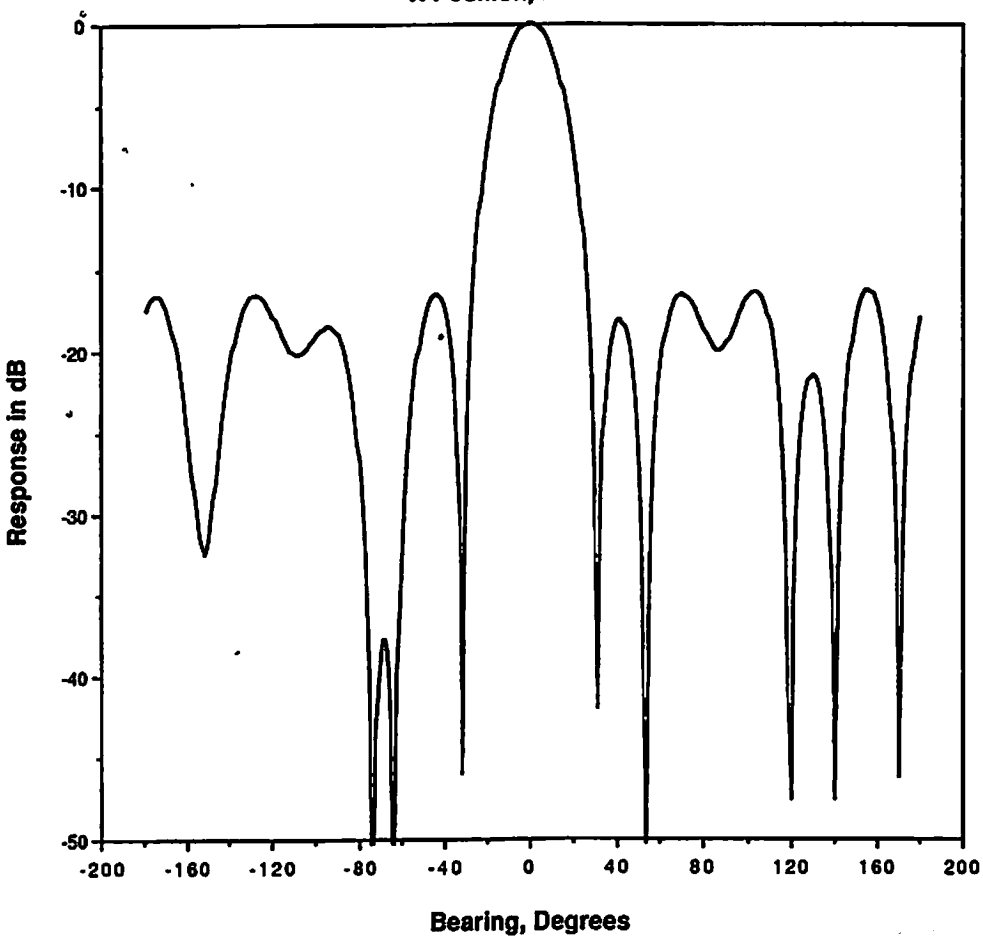


Figure 6. Unity shaded beam response for array of Figure 5.



a)



b)

Figure 7. Thinned array result using the l_q simplex search with mainlobe and maximum sidelobe constrained to match figure 5b. a) Element positions. b) Thinned and optimally shaded beam response.

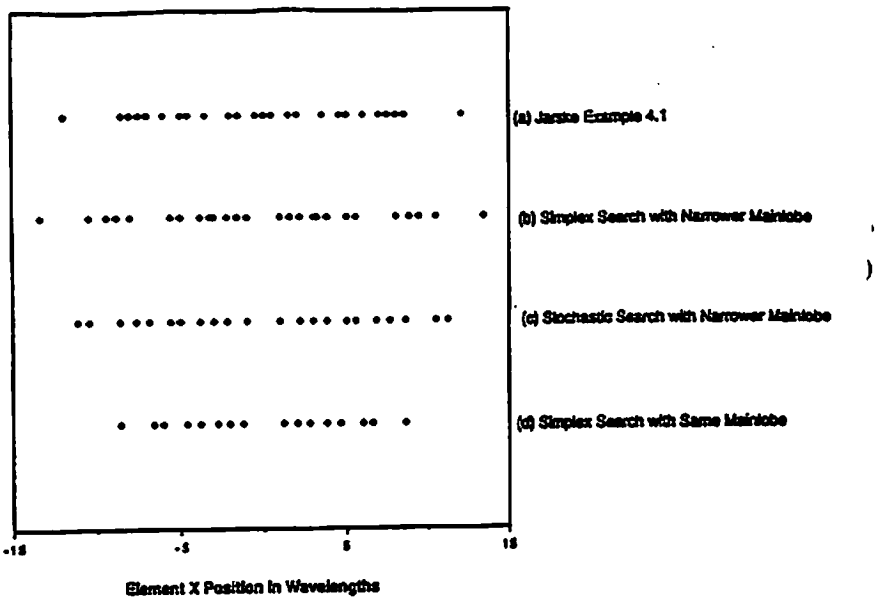


Figure 8. Element positions of four thinned array placement solutions. a) Jarske et. al. example 4.1 [25]. b) l_q simplex search with narrower mainlobe constraint than a. c) Stochastic search with narrow mainlobe. d) l_q simplex search with identical constraints to a. above.

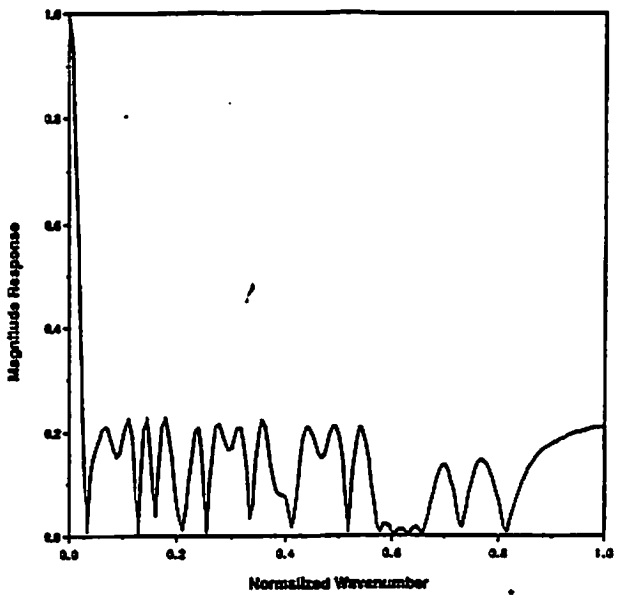


Figure 9. Beam magnitude response for array of Figure 7b.

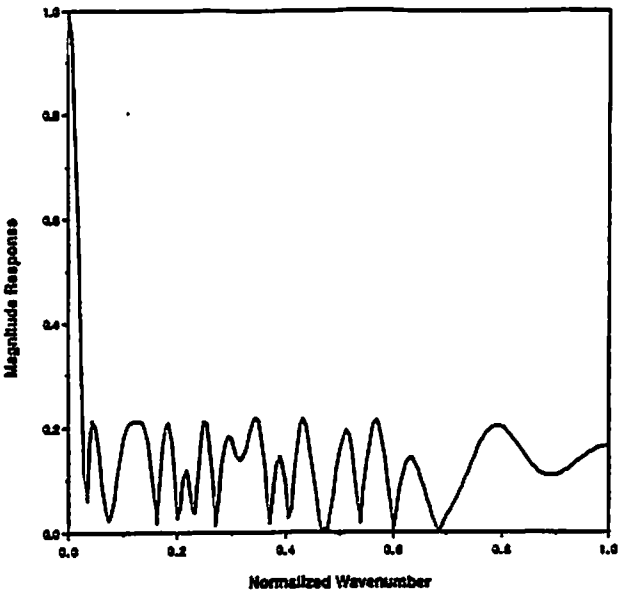


Figure 10. Beam magnitude response for array of Figure 7c.

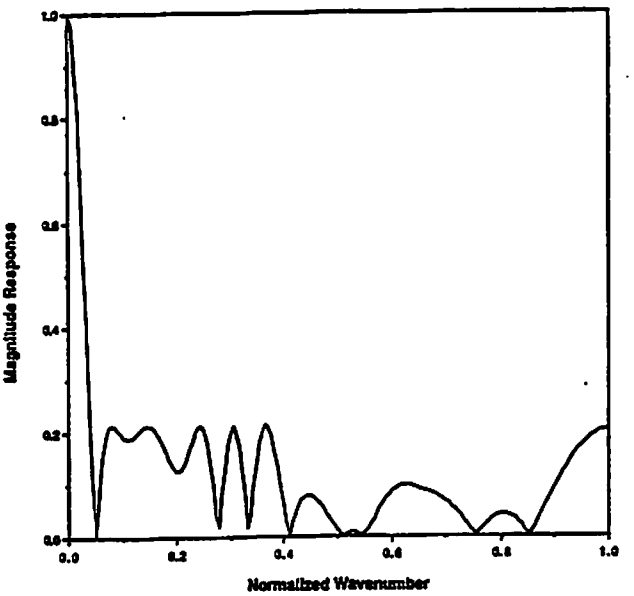


Figure 11. Beam magnitude response for array of Figure 7d.

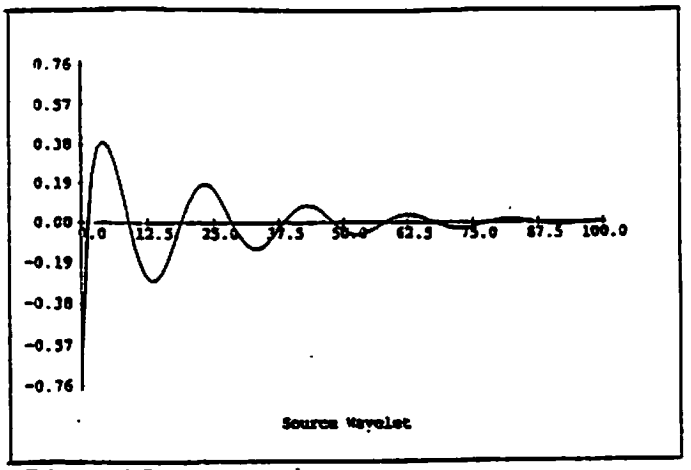


Fig. 12a

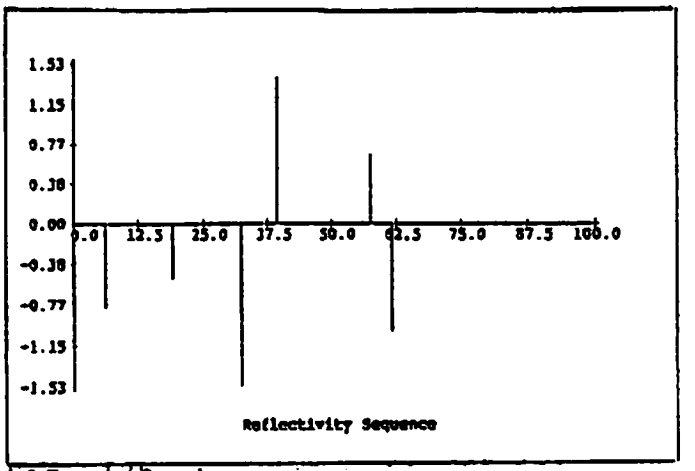


Fig. 12b

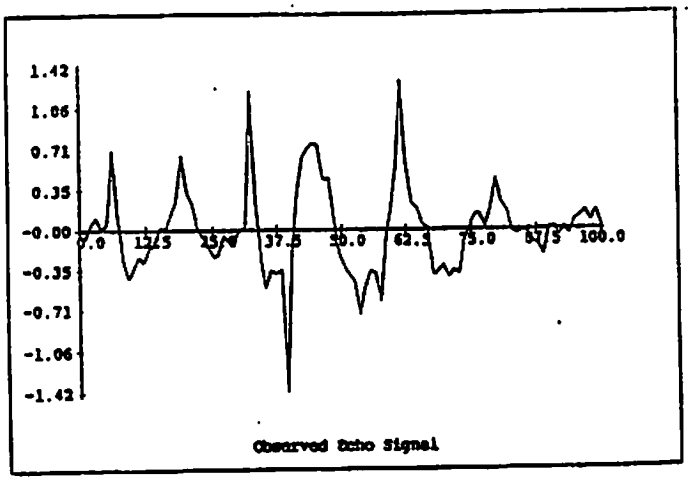


Fig. 13a

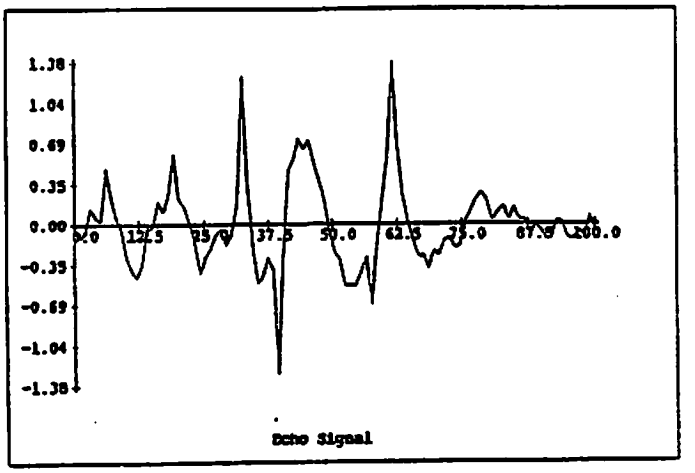


Fig 13b

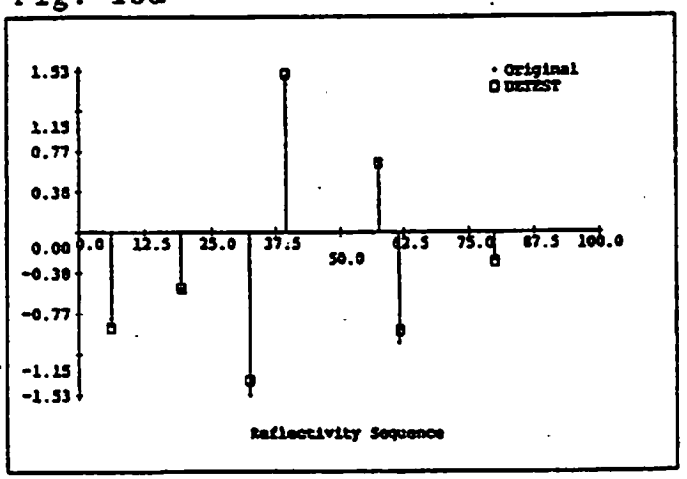


Fig 14a

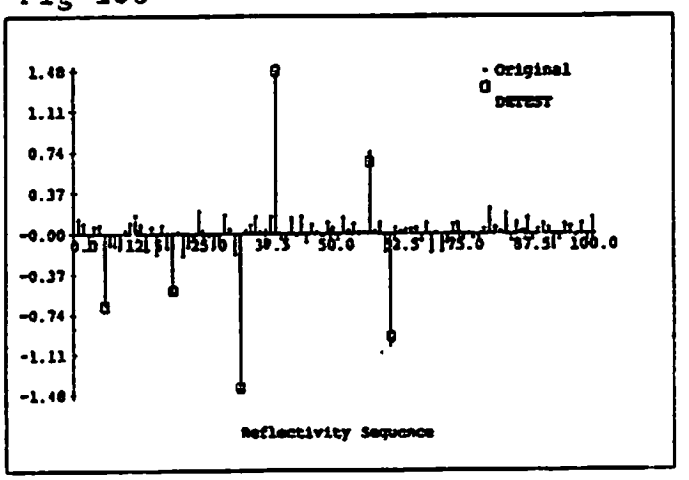


Fig. 14b

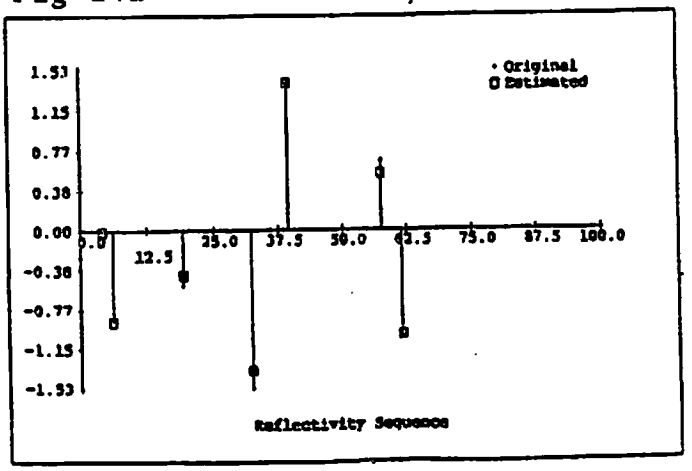


Fig. 15a

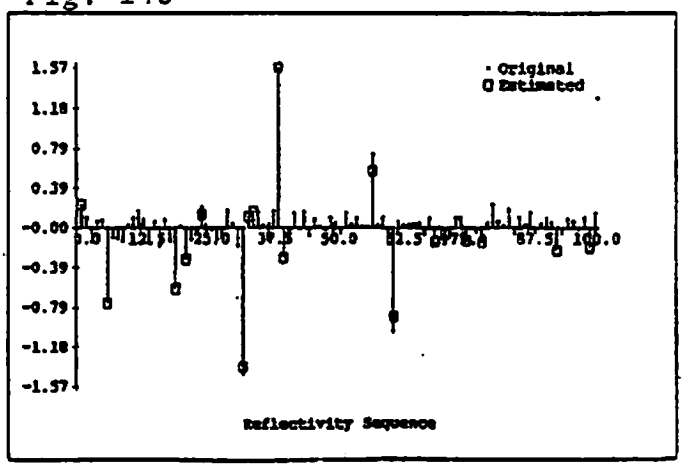


Fig. 15b

Captions:

Table 1. Optimal solutions to eqn (2) for the system of eqn (3) for various q.

Table 2. Comparison of the degeneracy of the feasible basic solutions to eqn (8).

Figure 1. Unit balls of the l_p norm for various p. Note that as p approaches 0 the unit ball approaches the axes.

Figure 2. Illustration of the cost, $g_p(x_i)$ associated with each of the four basic feasible solutions to the problem of eqn (4) for p values of $0<p\leq 1$. Note for $0<p\leq .613$, x_4 is of lowest cost, while for $.613<p\leq 1$, x_1 is of lowest cost.

Figure 3. An example of l_q optimization for various values of q. Solutions to equation (3) show minimum order results for $q>3.32$.

Figure 4. Graph of the simplex tableaus and their connections for the example in eqn (8). Each shaded area, DG_i, represents the degeneracy subgraph associated with basic feasible solution x_i. Graph node numbers indicate the corresponding unique tableaus.

Figure 5. Original 60 elements concentric ring array. Elements are omnidirectional and beams are formed in the plane of the array. Design is for 1 kHz acoustic operation in seawater.

Figure 6. Unity shaded beam response for array of Figure 5.

Figure 7. Thinned array result using the l_q simplex search with mainlobe and maximum sidelobe constrained to match figure 5b. a) Element positions. b) Thinned and optimally shaded beam response.

Figure 8. Element positions of four thinned array placement solutions. a) Jarske et. al. example 4.1 [25]. b) l_q simplex search with narrower mainlobe constraint than a. c) Stochastic search with narrow mainlobe. d) l_q simplex search with identical constraints to a. above.

Figure 9. Beam magnitude response for array of Figure 7b.

Figure 10. Beam magnitude response for array of Figure 7c.

Figure 11. Beam magnitude response for array of Figure 7d.

Figure 12. a) ARMA wavelet used in the seismic signal simulation. b) Reflectivity sequence used in the seismic data simulation.

Figure 13. a) Resulting signal synthesized from wavelet and reflectivity sequence of Figure 12. b) Signal with -10dB Gaussian noise added prior to convolution.

Figure 14. Deconvolution results using Mendel's optimal seismic deconvolution. a) Using data of Figure 13a. b) Using data of Figure 13b.

Figure 15. Deconvolutin results using nonlinear simplex algorithm. a) Using data of Figure 13a. b) Using data of Figure 13b.

Optimal Solutions				
q values:	Solution type:	x_1:	x_2:	x_3:
0	min-max	.91	.91	.91
.5	min energy	.98	.20	.98
$1 \leq q \leq 3.32$	linear program	1.00	0.00	1.00
$q > 3.32$	min order (max sparse)	0.00	10.00	0.00

Table 1: Optimal solutions to eqn (2) for the system of eqn (3) for various q.

BASIC FEASIBLE SOLUTION:						Order of Degeneracy	Tableaus in Degeneracy Subgraph
	x_1	x_2	x_3	x_4	x_5		
x_1	1	1	1	0	0	0	(1)
x_2	2	2	0	-2	0	0	(2)
x_3	0	0	0	0	6	2	(3),(5),(6),(7),(9)
x_4	0	0	2	2	0	1	(4),(8)

Table 2: Comparison of the degeneracy of the feasible basic solutions to eqn (8)

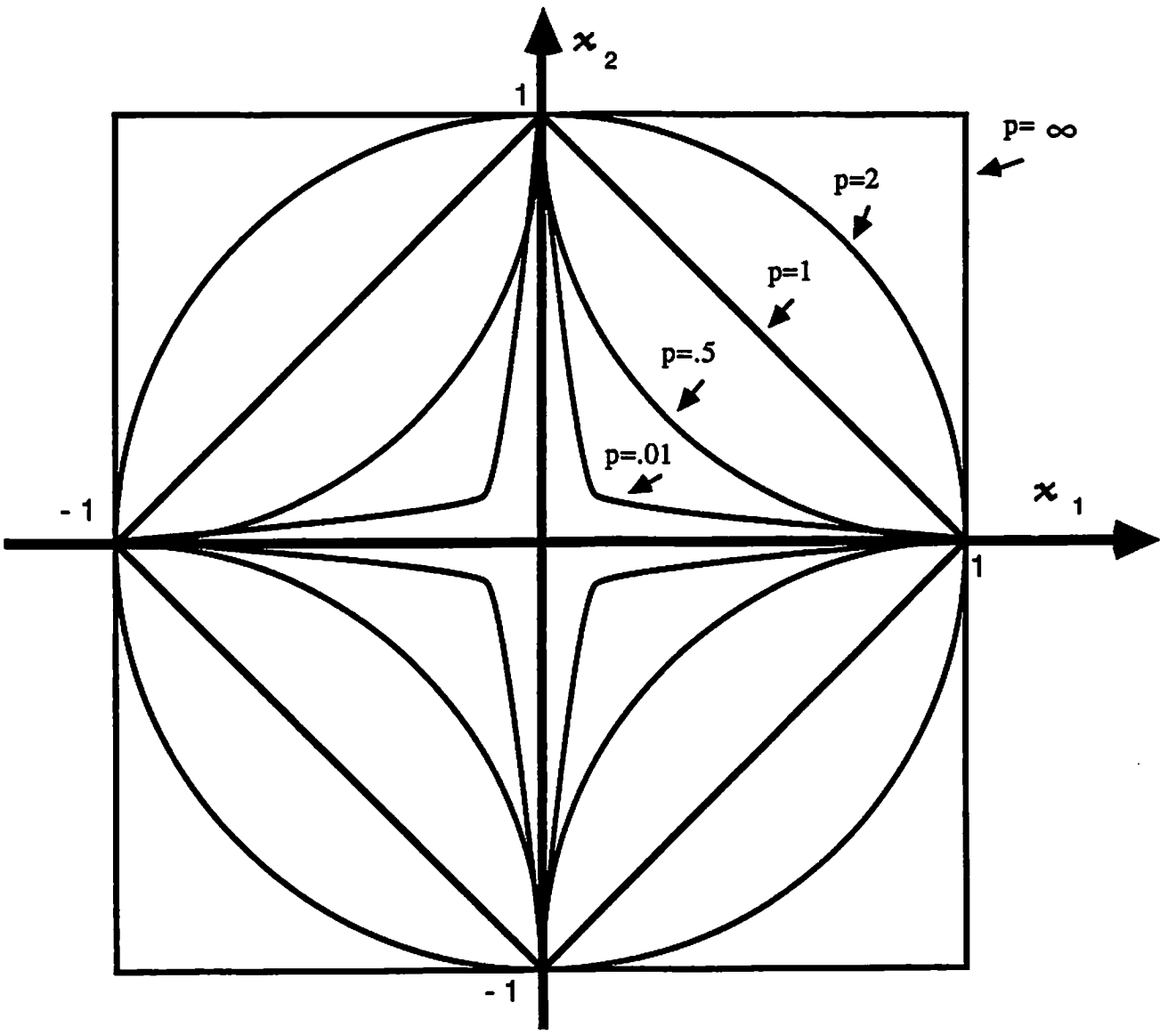


Figure 1. Unit balls of the l_p norm for various p. Note as p approaches 0 the unit ball approaches the axes.

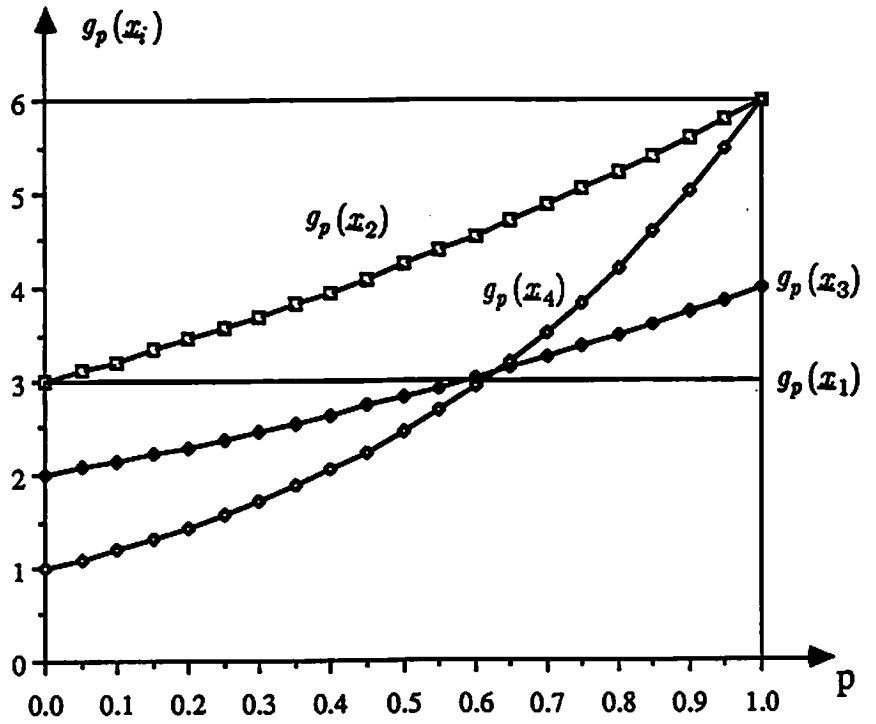


Figure 2. Illustration of the cost, $g_p(x_i)$ associated with each of the four basic feasible solutions to the problem of eqn (4) for p values of $0<p\leq 1$. Note for $0<p\leq .613$, x_4 is of lowest cost, while for $.613<p\leq 1$, x_1 is of lowest cost.

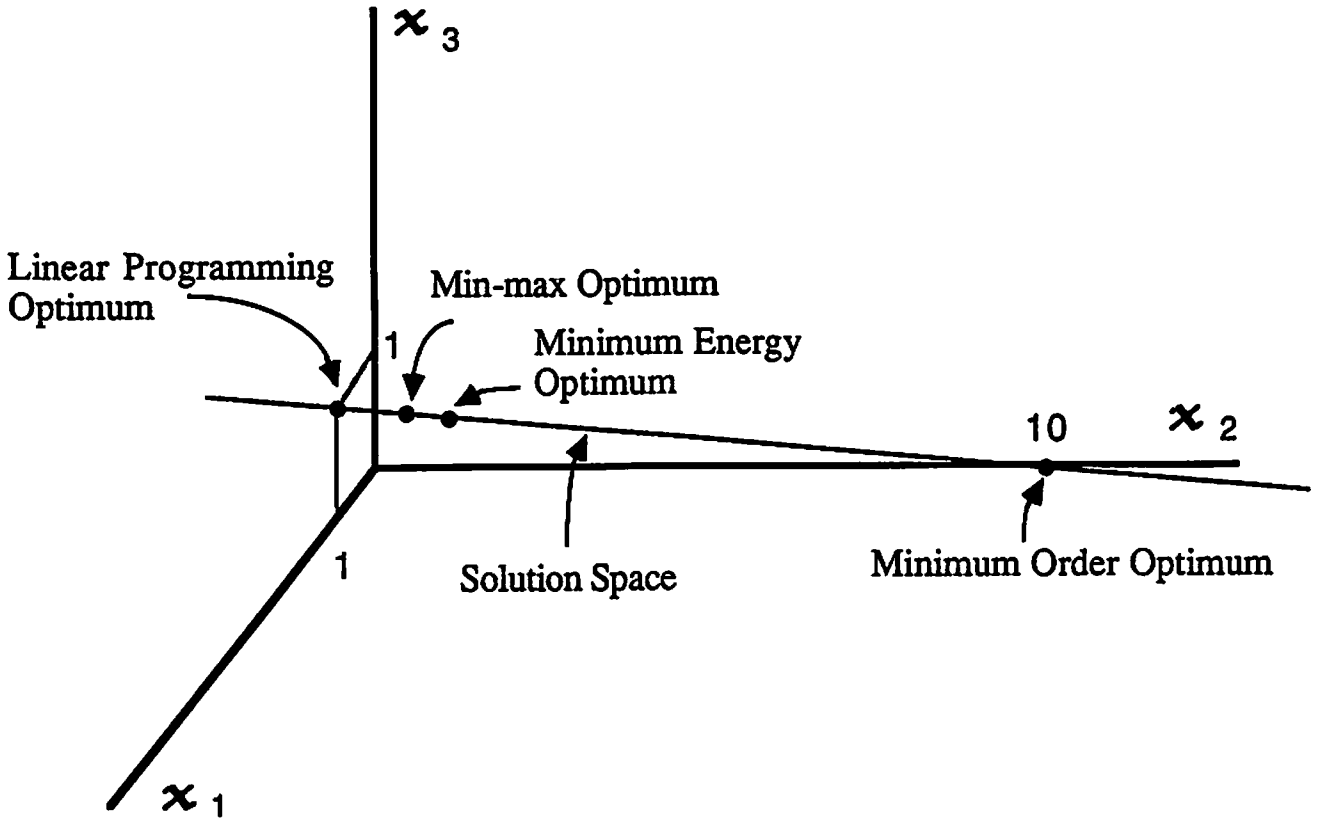


Figure 3. An example of l_q optimization for various values of q. Solutions to equation (3) show minimum order results for q>3.32

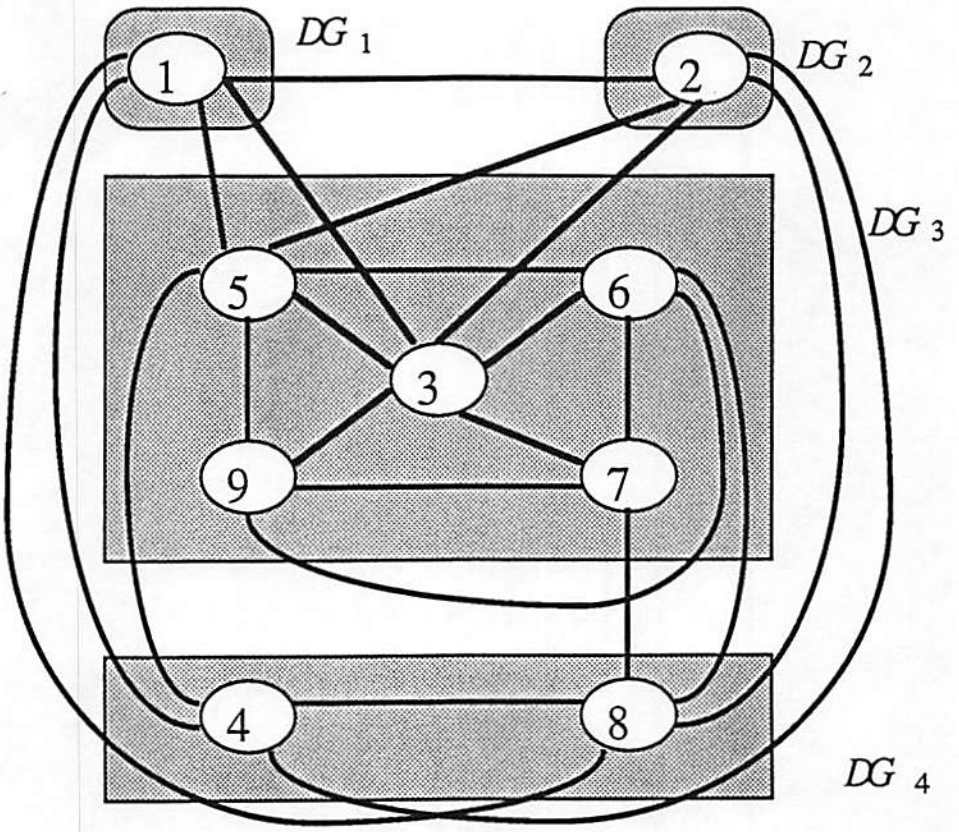


Figure 4: Graph of the simplex tableaus and their connections for the example in eqn (8). Each shaded area, DG_i , represents the degeneracy subgraph associated with basic feasible solution Xi. Graph node numbers indicate the corresponding unique tableaus.

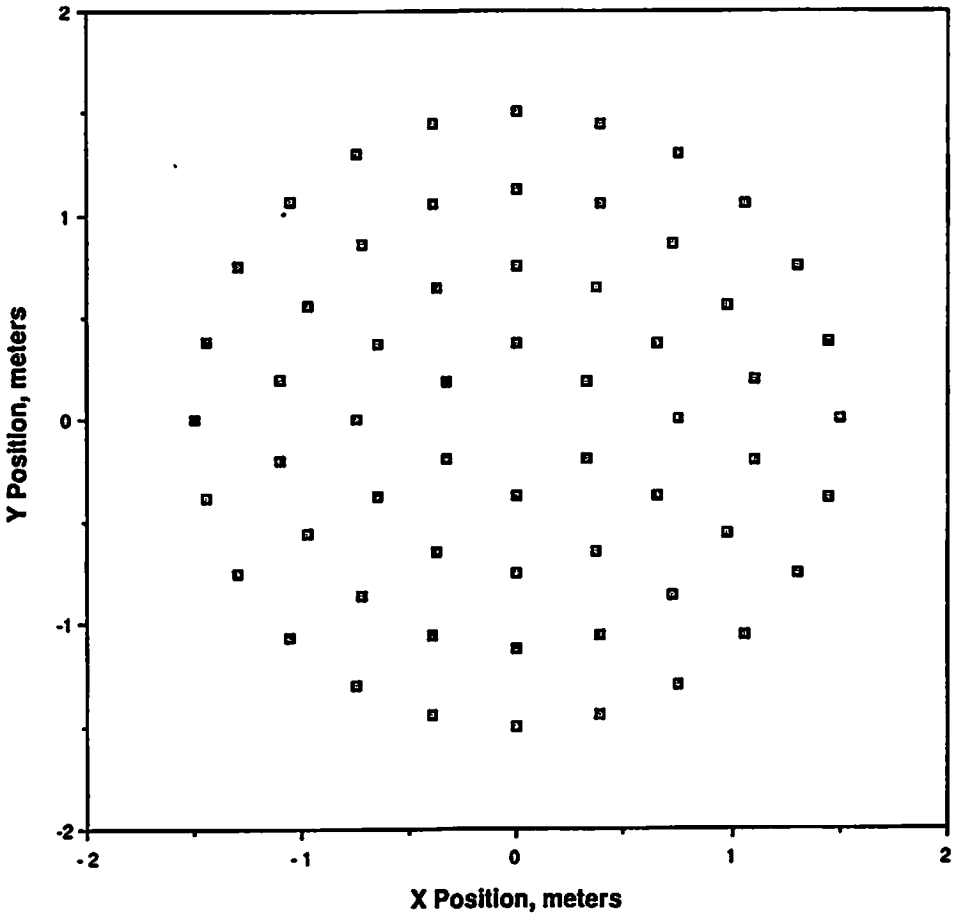


Figure 5. Original 60 elements concentric ring array. Elements are omnidirectional and beams are formed in the plane of the array. Design is for 1 kHz acoustic operation in seawater.

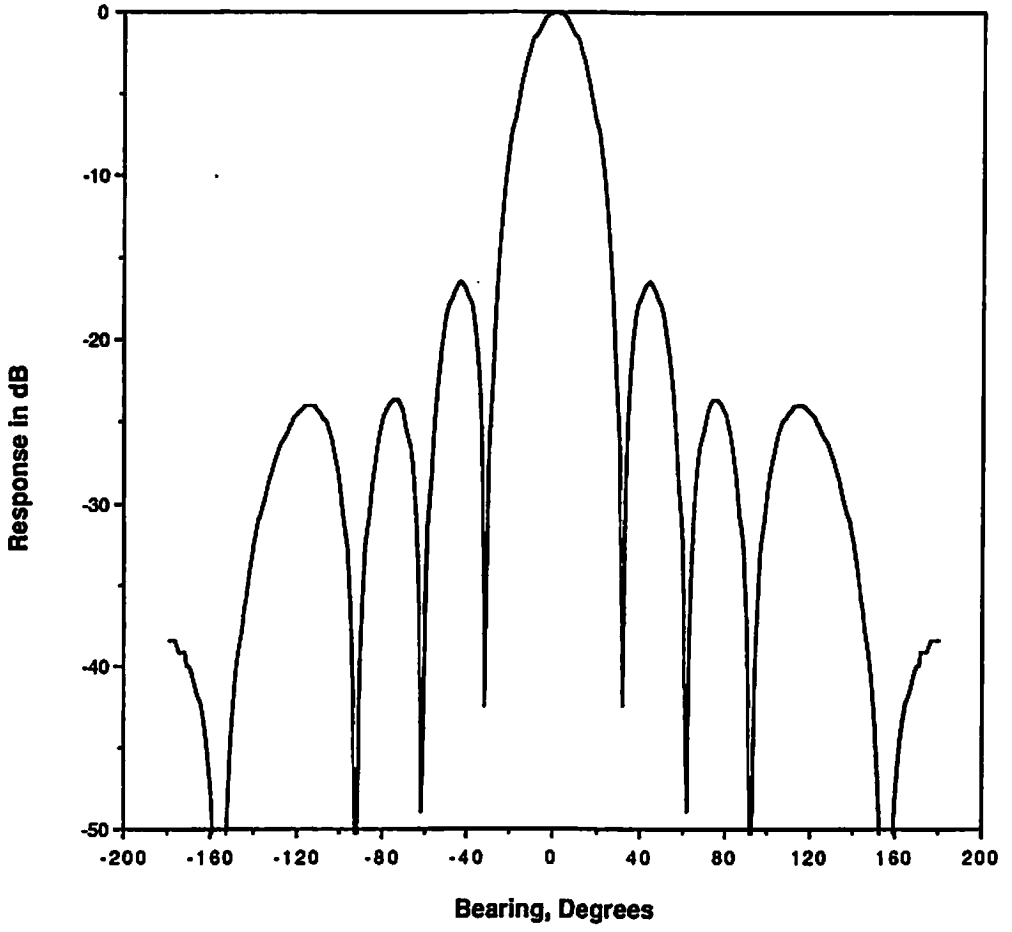
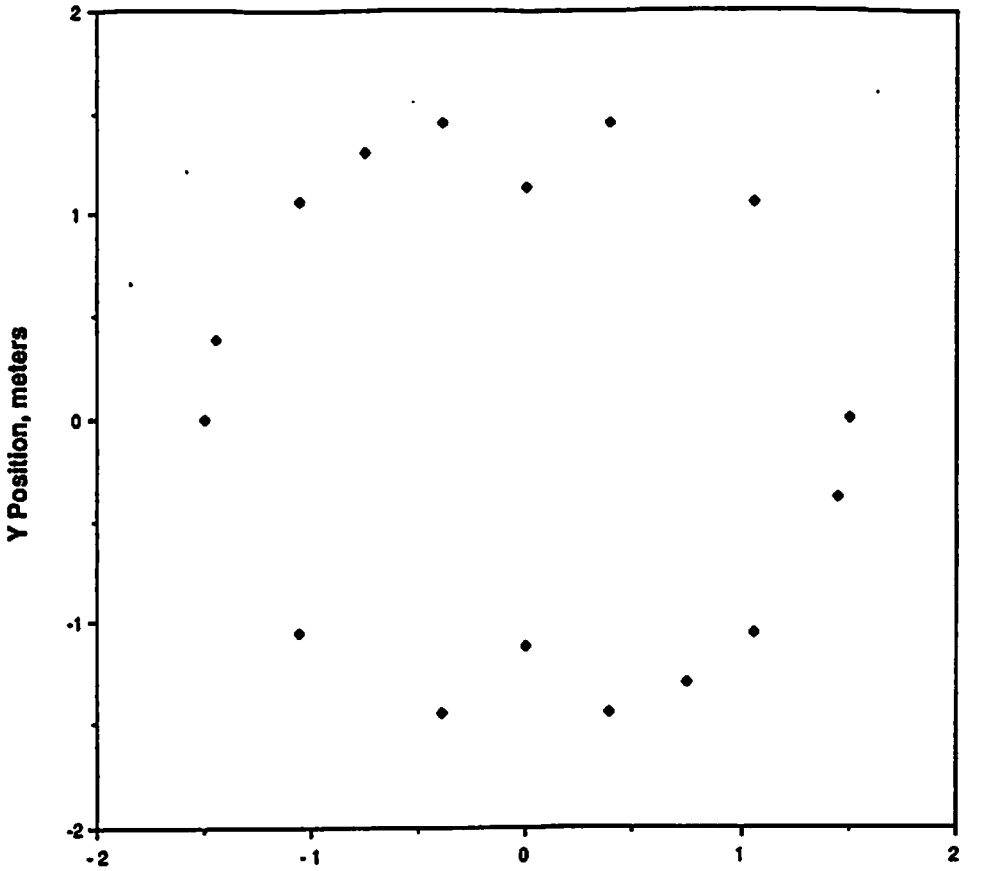
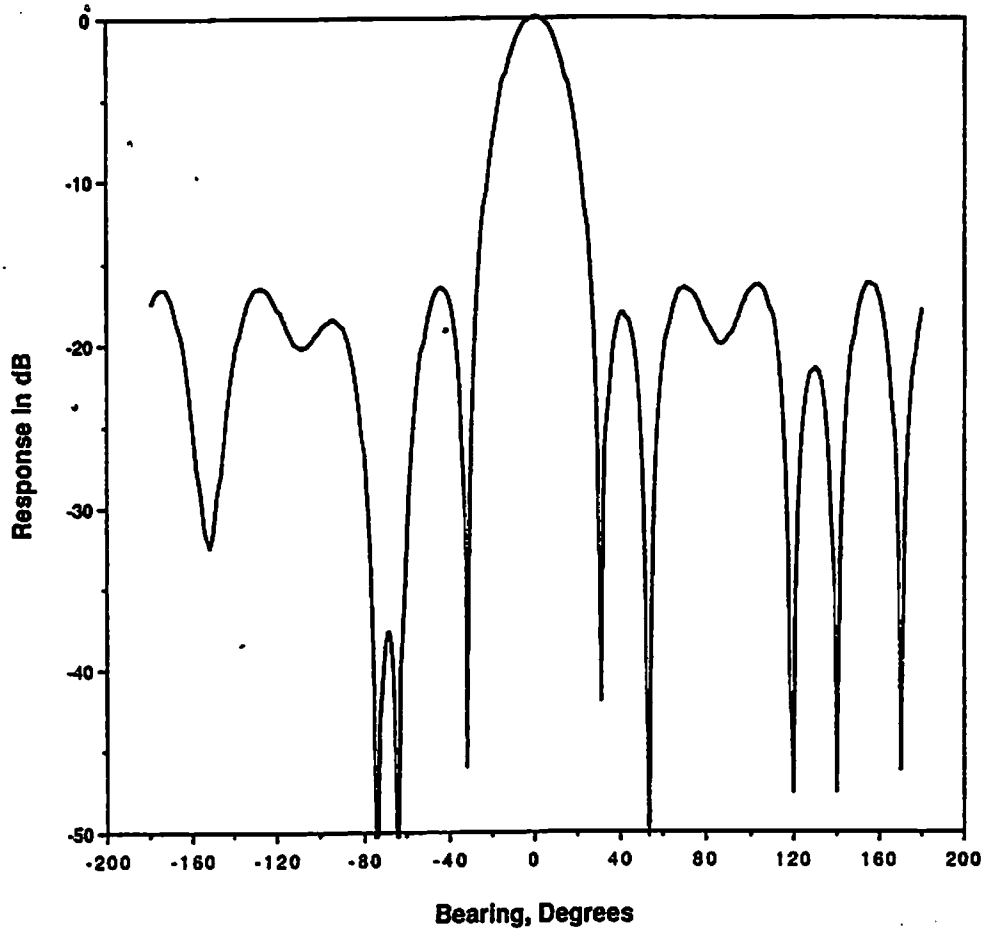


Figure 6. Unity shaded beam response for array of Figure 5.



a)



b)

Figure 7. Thinned array result using the l_q simplex search with mainlobe and maximum sidelobe constrained to match figure 5b. a) Element positions. b) Thinned and optimally shaded beam response.

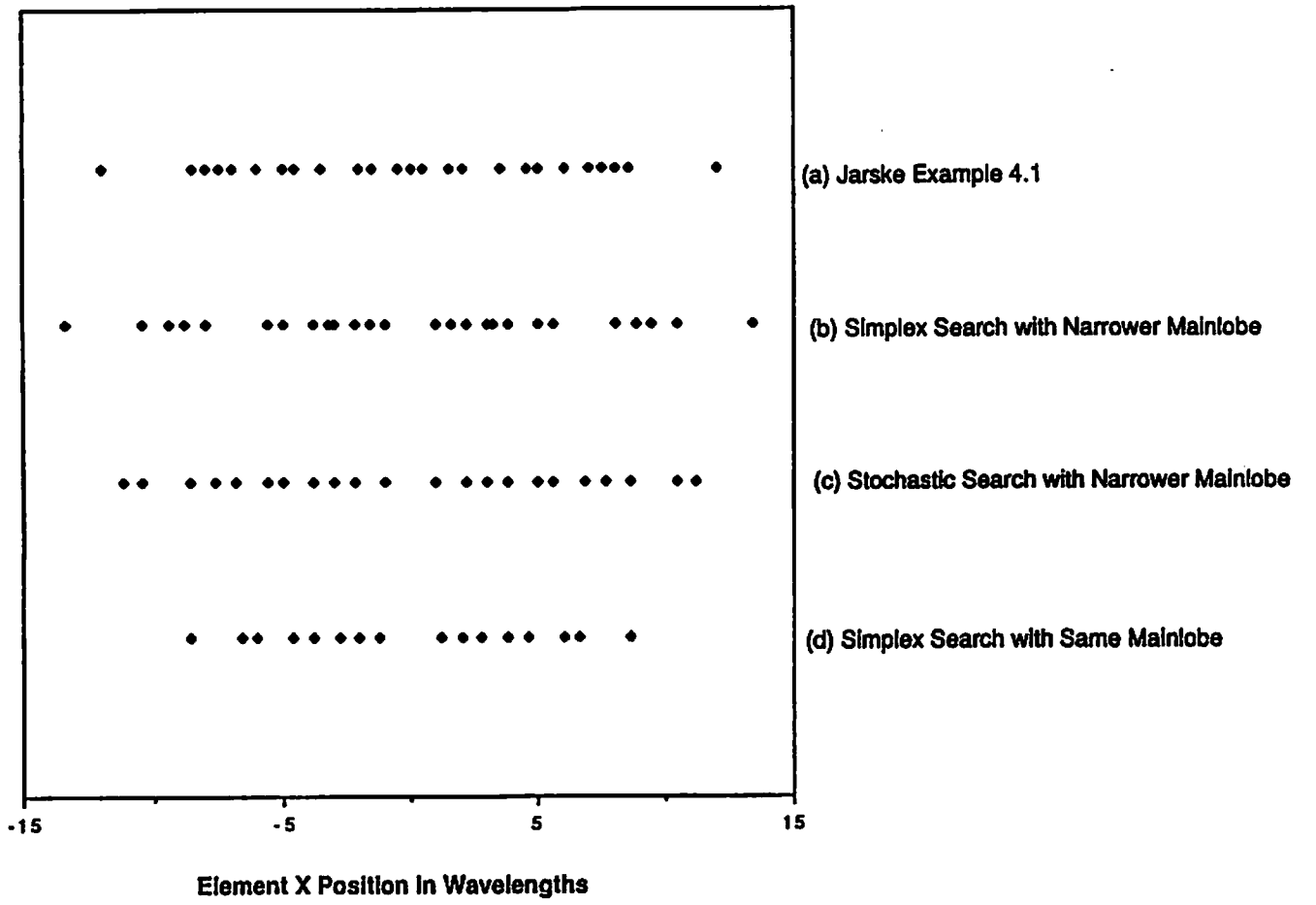


Figure 8. Element positions of four thinned array placement solutions. a) Jarske et. al. example 4.1 [25]. b) l_q simplex search with narrower mainlobe constraint than a. c) Stochastic search with narrow mainlobe. d) l_q simplex search with identical constraints to a. above.

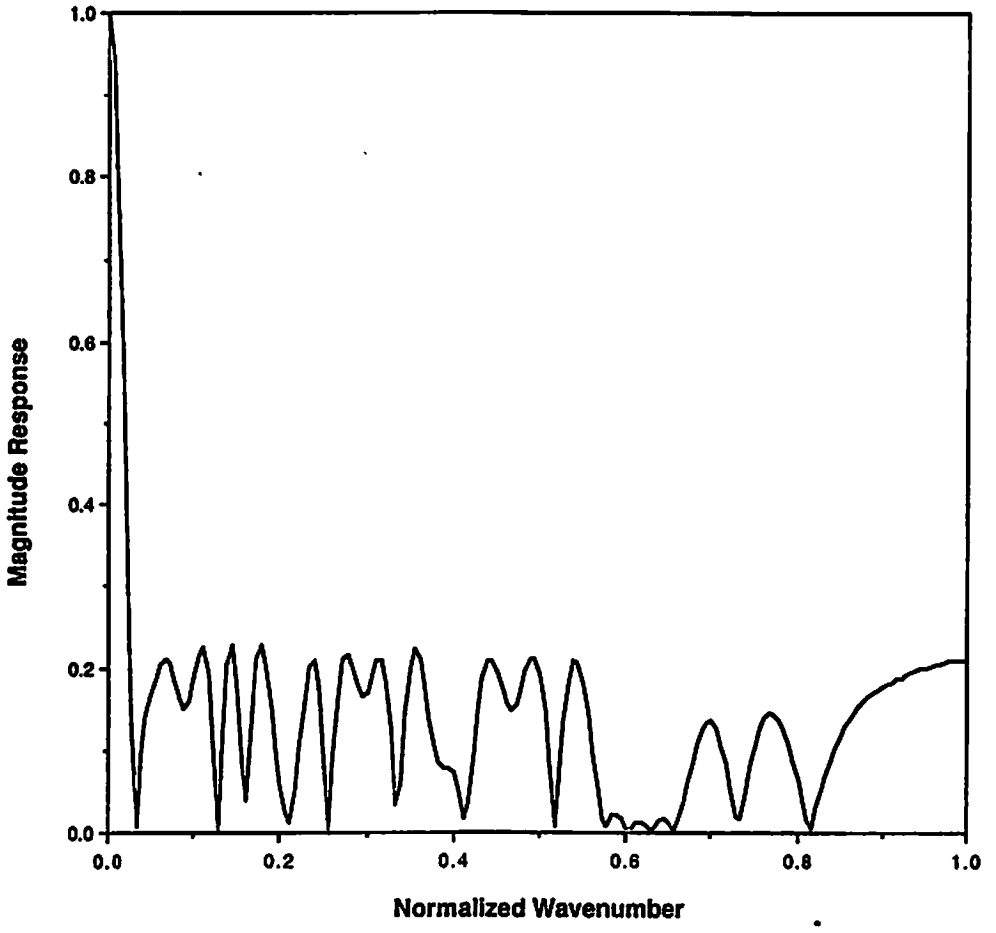


Figure 9. Beam magnitude response for array of Figure 7b.

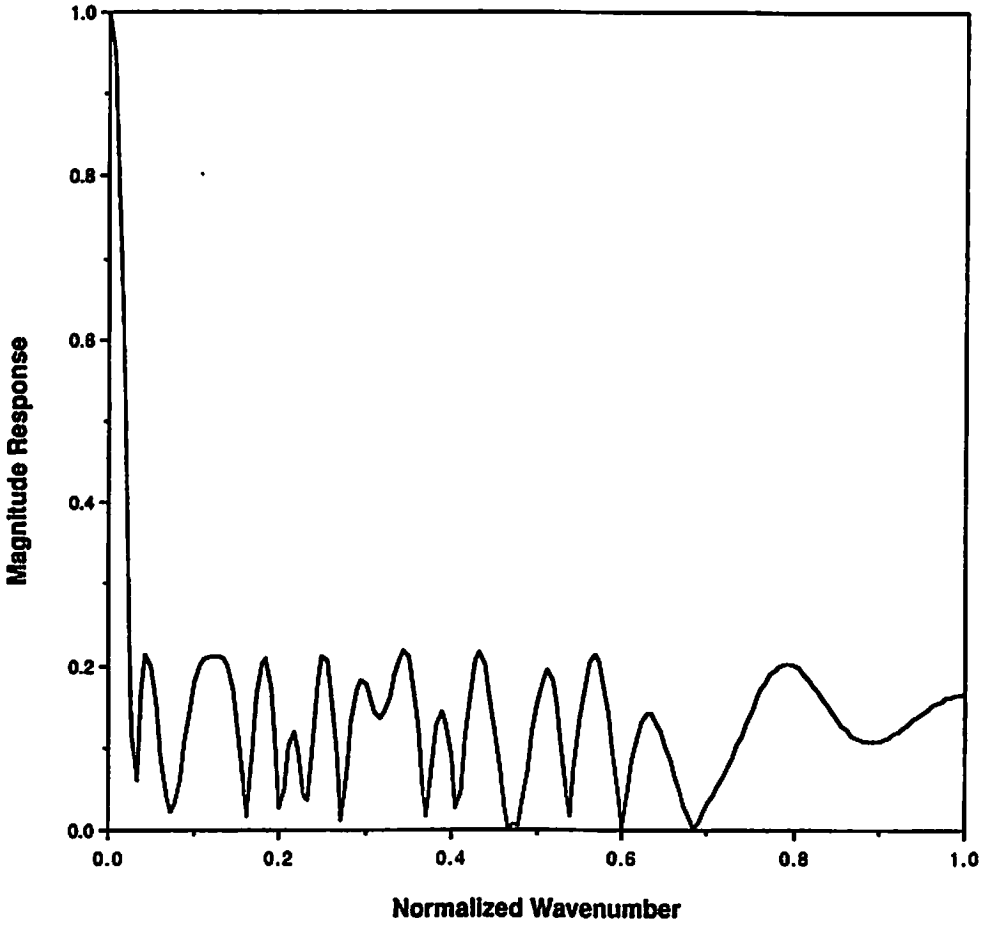


Figure 10. Beam magnitude response for array of Figure 7c.

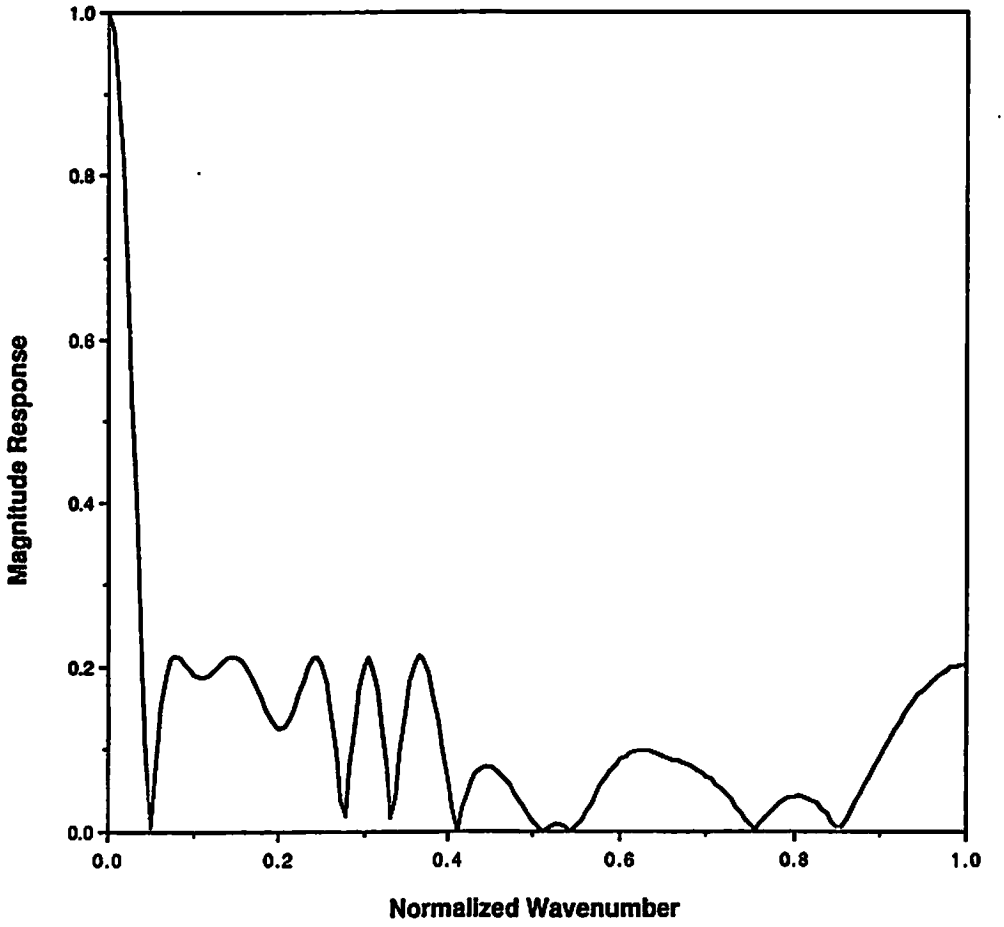


Figure 11. Beam magnitude response for array of Figure 7d.

Optimal Solutions				
q values:	Solution type:	x_1:	x_2:	x_3:
0	min-max	.91	.91	.91
.5	min energy	.98	.20	.98
$1 \leq q \leq 3.32$	linear program	1.00	0.00	1.00
$q > 3.32$	min order (max sparse)	0.00	10.00	0.00

Table 1: Optimal solutions to eqn (2) for the system of eqn (3) for various q.

BASIC FEASIBLE SOLUTION:						Order of Degeneracy	Tableaus in Degeneracy Subgraph
	x_1	x_2	x_3	x_4	x_5		
x_1	1	1	1	0	0	0	(1)
x_2	2	2	0	-2	0	0	(2)
x_3	0	0	0	0	6	2	(3),(5),(6),(7),(9)
x_4	0	0	2	2	0	1	(4),(8)

Table 2: Comparison of the degeneracy of the feasible basic solutions to eqn (8)

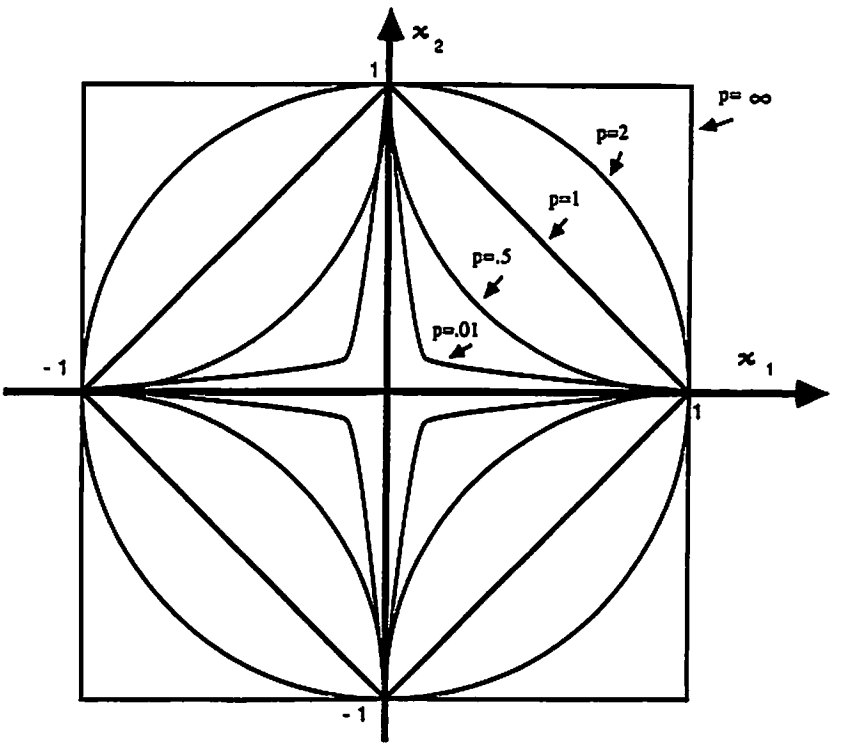


Figure 1. Unit balls of the l_p norm for various p. Note as p approaches 0 the unit ball approaches the axes.

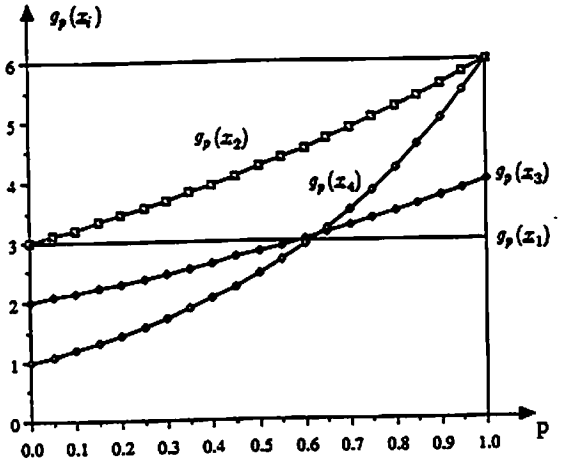


Figure 2. Illustration of the cost, $g_p(x_i)$ associated with each of the four basic feasible solutions to the problem of eqn (4) for p values of $0<p\leq 1$. Note for $0<p\leq .613$, x_4 is of lowest cost, while for $.613<p\leq 1$, x_1 is of lowest cost.

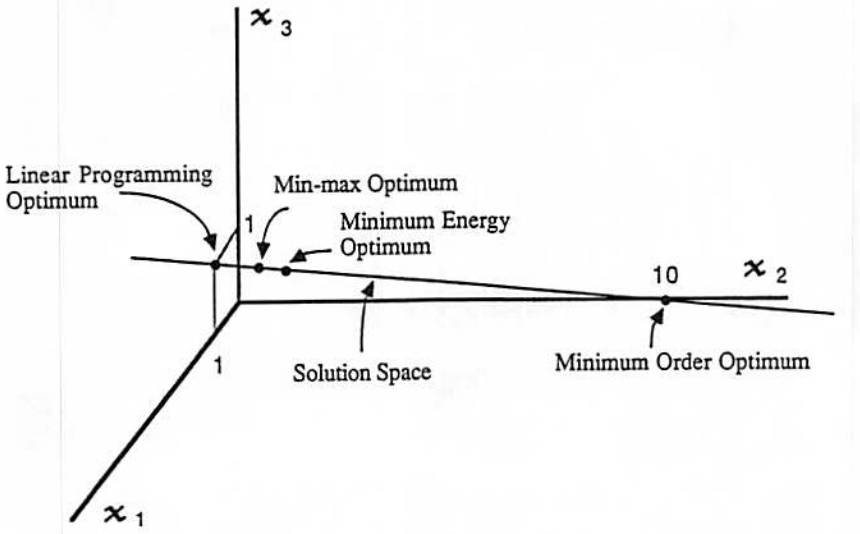


Figure 3. An example of 1_q optimization for various values of q. Solutions to equation (3) show minimum order results for q>3.32

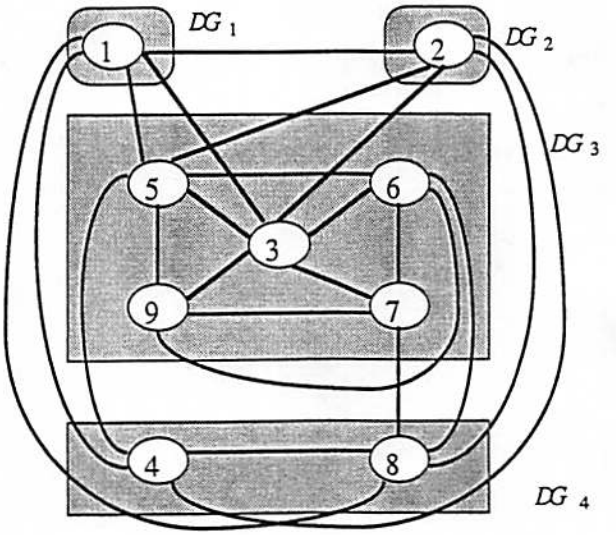


Figure 4: Graph of the simplex tableaus and their connections for the example in eqn (8). Each shaded area, DG_i , represents the degeneracy subgraph associated with basic feasible solution Xi. Graph node numbers indicate the corresponding unique tableaus.

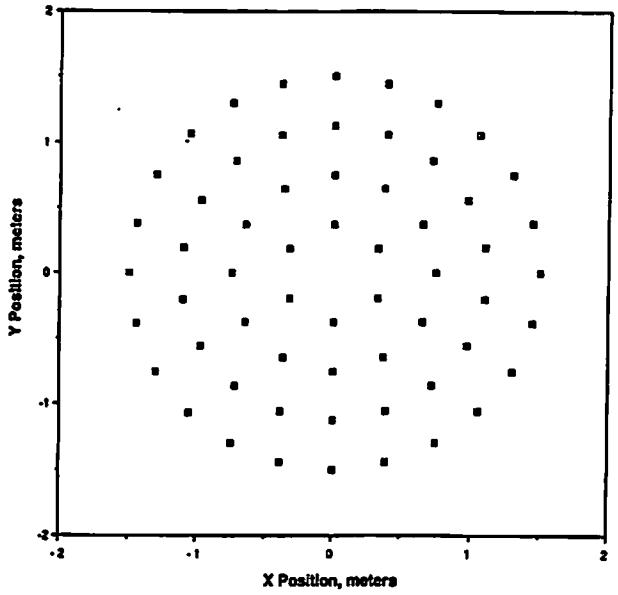


Figure 5. Original 60 elements concentric ring array. Elements are omnidirectional and beams are formed in the plane of the array. Design is for 1 kHz acoustic operation in seawater.

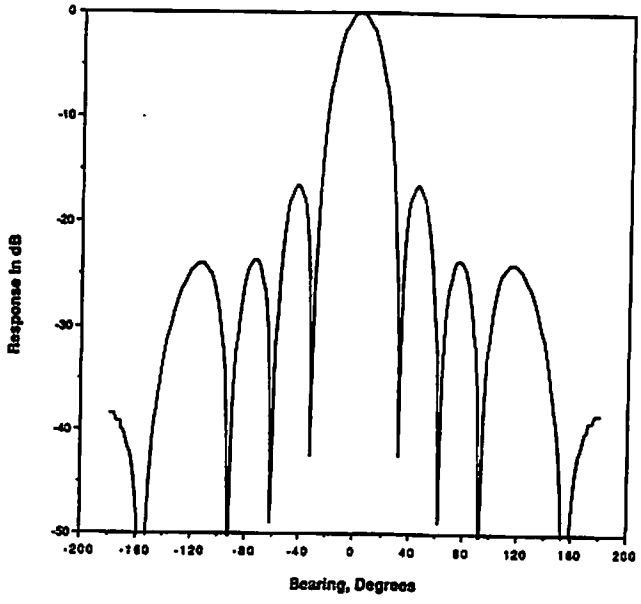
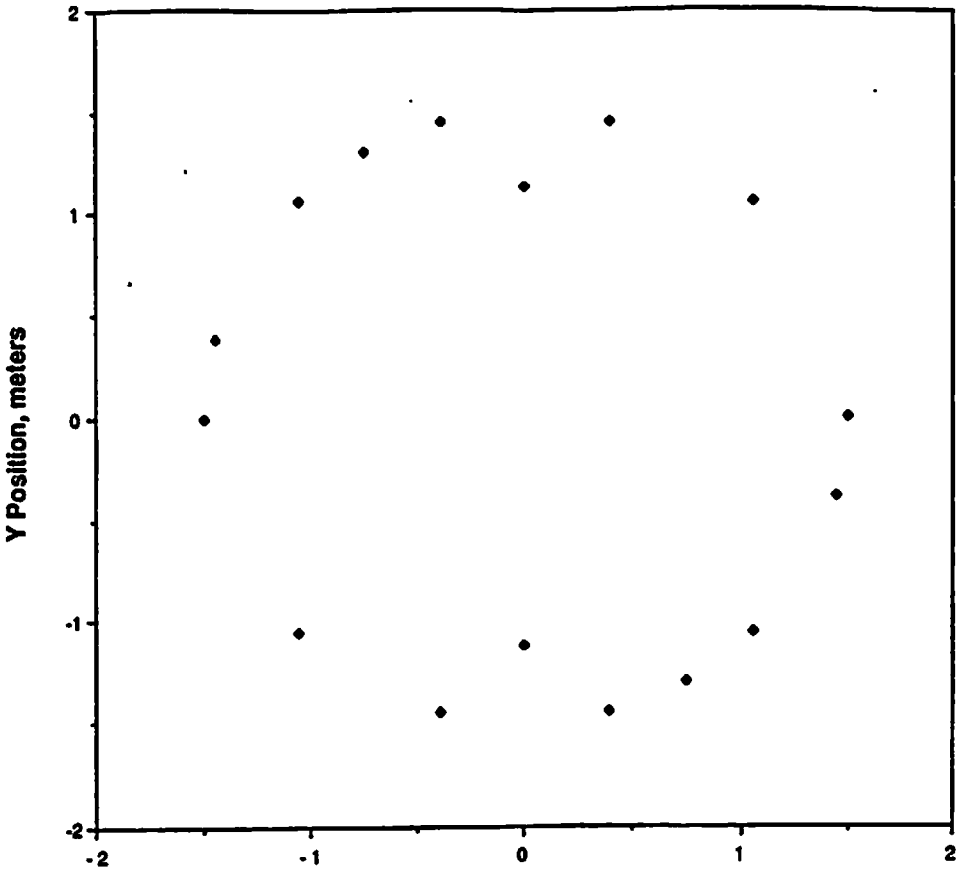
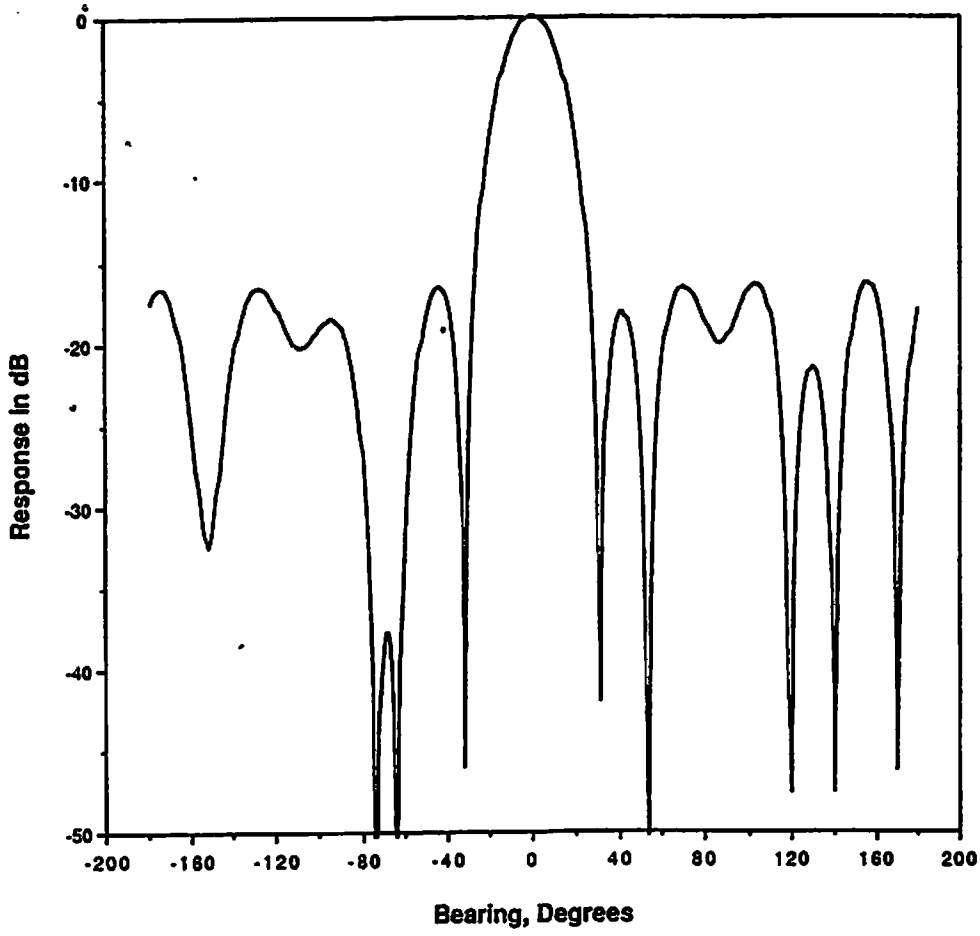


Figure 6. Unity shaded beam response for array of Figure 5.



a)



b)

Figure 7. Thinned array result using the l_q simplex search with mainlobe and maximum sidelobe constrained to match figure 5b. a) Element positions. b) Thinned and optimally shaded beam response.

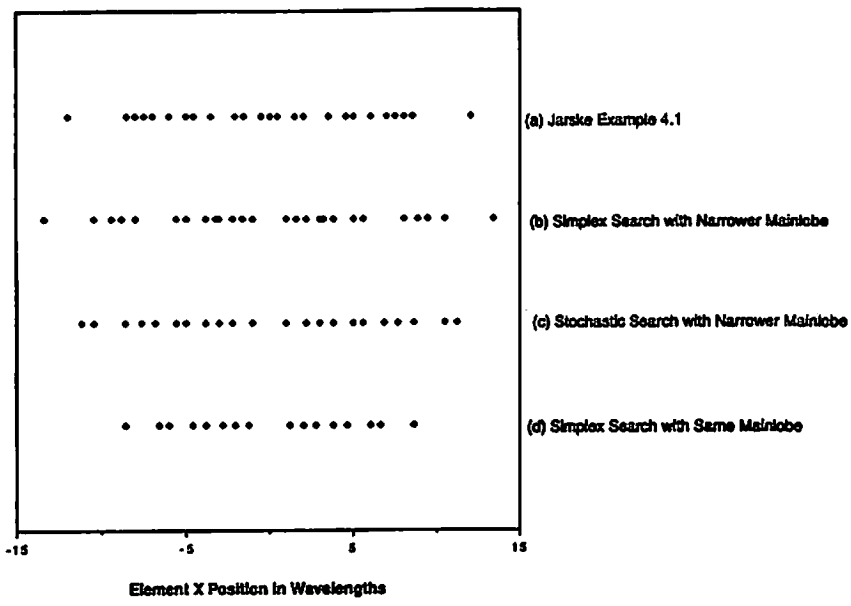


Figure 8. Element positions of four thinned array placement solutions. a) Jarske et. al. example 4.1 [25]. b) l_q simplex search with narrower mainlobe constraint than a. c) Stochastic search with narrow mainlobe. d) l_q simplex search with identical constraints to a. above.

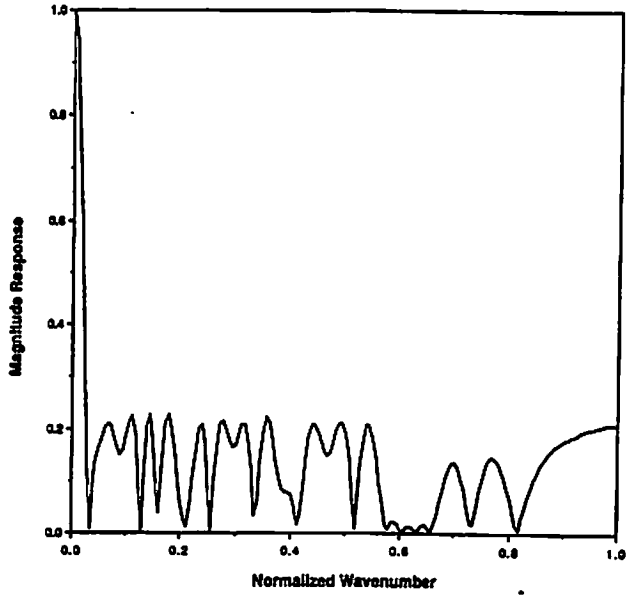


Figure 9. Beam magnitude response for array of Figure 7b.

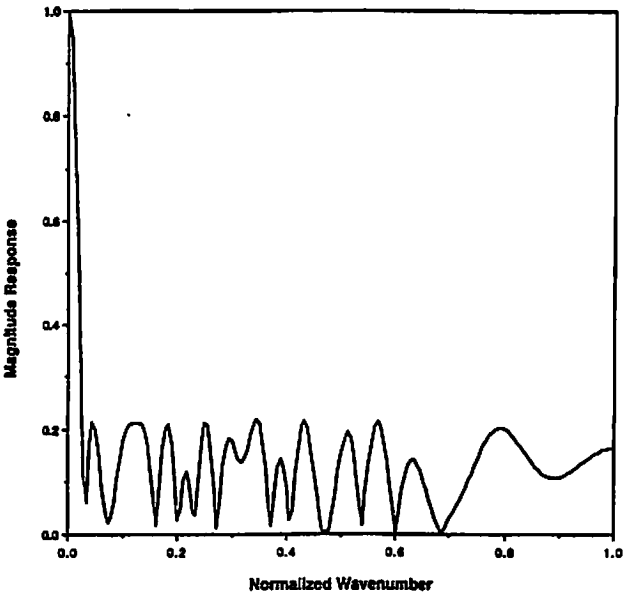


Figure 10. Beam magnitude response for array of Figure 7c.

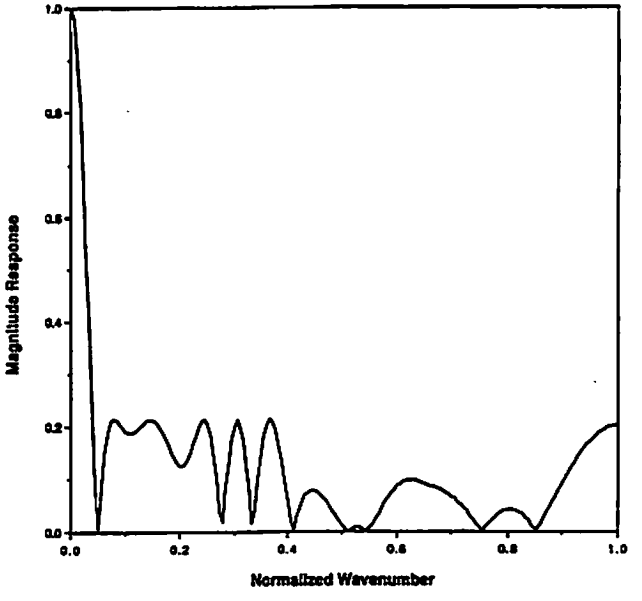


Figure 11. Beam magnitude response for array of Figure 7d.