On the Computational Complexity of 2D Maximum-Likelihood Sequence Detection

Erik Ordentlich, Ron M. Roth
HP Laboratories Palo Alto
HPL-2006-69
April 17, 2006*

two-dimensional, maximum-likelihood, sequence detection, intersymbol interference, multiuser detection, Viterbi algorithm, NP completeness

We consider a two-dimensional version of the classical maximum-likelihood sequence detection problem for a binary antipodal signal that is corrupted by linear intersymbol interference (ISI) and then passed through a memoryless channel. For one-dimensional signals and fixed ISI, this detection problem is well-known to be efficiently solved using the Viterbi algorithm. We show that the two-dimensional version is, in general, intractable, in the sense that a decision formulation of the problem is NP complete for a certain fixed two-dimensional ISI and memoryless channel with errors and erasures. We also extend the result to the additive white Gaussian noise channel and the same fixed ISI as in the previous case. Finally, we show how our proof of NP completeness for the two-dimensional case can also be used to prove the NP completeness of multiuser detection under a Toeplitz constraint, shown in [4] to be equivalent to a variant of one-dimensional maximum likelihood sequence detection when the ISI is not fixed, thereby proving Conjecture 1 in [4].

* Internal Accession Date Only
Presented in part at the ITA Workshop, 8 February 2006, San Diego, CA, USA
Approved for External Publication

On the Computational Complexity of 2D Maximum-Likelihood Sequence Detection

Erik Ordentlich
1501 Page Mill Road
Hewlett-Packard Laboratories
Palo Alto, CA 94304, U.S.A.
eord@hpl.hp.com

Ron M. Roth*
Computer Science Department
Technion
Haifa 32000, Israel
ronny@cs.technion.ac.il

April 13, 2006

Abstract

We consider a two-dimensional version of the classical maximum-likelihood sequence detection problem for a binary antipodal signal that is corrupted by linear intersymbol interference (ISI) and then passed through a memoryless channel. For one-dimensional signals and fixed ISI, this detection problem is well-known to be efficiently solved using the Viterbi algorithm. We show that the two-dimensional version is, in general, intractable, in the sense that a decision formulation of the problem is NP complete for a certain fixed two-dimensional ISI and memoryless channel with errors and erasures. We also extend the result to the additive white Gaussian noise channel and the same fixed ISI as in the previous case. Finally, we show how our proof of NP completeness for the two-dimensional case can also be used to prove the NP completeness of multiuser detection under a Toeplitz constraint, shown in [4] to be equivalent to a variant of one-dimensional maximum likelihood sequence detection when the ISI is not fixed, thereby proving Conjecture 1 in [4].

Keywords: Two-dimensional, maximum-likelihood, sequence detection, intersymbol interference, multiuser detection, Viterbi algorithm, NP completeness.

1 Introduction

Consider a two-dimensional (2D) communication system in which an $n \times m$ input signal array $\{x_{i,j} \in \{-1,1\} : 1 \leq i \leq n, 1 \leq j \leq m\}$ is corrupted by 2D linear intersymbol interference (ISI) cascaded with a memoryless channel. Specifically, let $\{h_{i,j} \in \mathbb{R} : -N \leq i,j \leq N\}$ denote the real-valued coefficients of a finite-extent 2D linear filter, and let $W(y|u)$ denote a conditional probability distribution defining the memoryless channel, where $u \in \mathcal{U} \triangleq \{\sum_{-N \leq i,j \leq N} h_{i,j} z_{i,j} : z_{i,j} \in \{-1,1\}\}$ and $y \in \mathcal{Y}$ where $\mathcal{Y}$ is some channel output alphabet. The conditional distribution of the output $\{y_{i,j}\}$ given the input $\{x_{i,j}\}$

*Work done while visiting Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, USA.

of the 2D communication system under consideration is then

$$P^{(n,m)}(\{y_{i,j}\}|\{x_{i,j}\}) = \prod_{\substack{i,j:1\le i\le n,\\ 1\le j\le m}} W(y_{i,j}|\sum_{r,s:-N\le r,s\le N} h_{r,s}x_{i+r,j+s}) \tag{1}$$

where $x_{i,j}$ is set to 1 for $(i,j) \notin \{1,\ldots,n\} \times \{1,\ldots,m\}$.

Given an output $\{y_{i,j}\}$, we are interested in performing 2D maximum-likelihood sequence detection (MLSD), or, formally, in computing

$$\arg\max_{\{x_{i,j}\}} P^{(n,m)}(\{y_{i,j}\}|\{x_{i,j}\}).$$

We are specifically interested in how the complexity of this computation behaves in the parameters n and m, for fixed $\{h_{i,j}\}$. In the case that, say, m is also fixed, or in the case that there is an i_0 (or j_0) such that $h_{i,j} = 0$ for $i \neq i_0$ (or for $j \neq j_0$), the resulting 2D MLSD problem collapses to a 1D problem, which is well-known to be efficiently solved in polynomial time (actually linear time) by the Viterbi algorithm. In all other cases, it has generally been thought, though, to our knowledge, not explicitly established, that the problem is intractable [1].

In this work, we formally establish this intractability for the specific "L–shaped" 2D ISI filter with

$$h_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \{(0,0),(-1,0),(0,-1)\}, \\ 0 & \text{otherwise,} \end{cases}$$

and channel W having inputs $\mathcal{U} = \{-3,-1,1,3\}$, outputs $\mathcal{Y} = \{-3,-1,1,3,e\}$, and transition probabilities

$$W(y|u) = \begin{cases} \delta & \text{if } y = e, \\ \epsilon & \text{if } y \neq u \text{ and } y \neq e, \\ 1-3\epsilon-\delta & \text{if } y = u. \end{cases} \tag{2}$$

Thus, the channel W induces erasures (output symbol e) and errors in a symmetrically-distributed fashion.

It is easy to see that when $\epsilon < (1-\delta)/4$, maximum-likelihood sequence detection for the above ISI and channel model is equivalent to

$$\arg\min_{\{x_{i,j}\}} \sum_{\substack{1\le i\le m\\ 1\le j\le n}} d(y_{i,j}, x_{i,j} + x_{i-1,j} + x_{i,j-1}), \tag{3}$$

where $d(\cdot,\cdot)$ is a distance metric defined as

$$d(y,u) = \begin{cases} 0 & \text{if } y = e \text{ or } y = u, \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Erasures, in particular, have zero distance with respect to all ISI outputs. Clearly, if (3) were tractable so would the following decision version (where we set $n = m$):

$\ldots$	$3/1$	$3/1$	$3/1$	$3/1$	$3/1$	$3/1$	$\ldots$
$\ldots$	$1/x$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$\ldots$
			$e/1$	$1/x$	$e/$		
			$3/1$	$1/\overline{x}$			
			$3/1$	$1/x$			
			$3/1$	$1/\overline{x}$			
			$3/1$	$1/x$			
			$\vdots$	$\vdots$			

Figure 1: Wires: downward T–connection.

Problem 1.1 *Given an $n \times n$ channel output array $\{y_{i,j}\}$, does*

$$\min_{\{x_{i,j}\}} \sum_{1 \le i,j \le n} d(y_{i,j}, x_{i,j} + x_{i-1,j} + x_{i,j-1}) = 0 \tag{5}$$

hold, where the minimization is over all $n \times n$ arrays of $\{+1, -1\}$?

Since the condition (5) can be checked in $O(n^2)$ operations for any input array $\{x_{i,j}\}$, it follows that Problem 1.1 is in the complexity class NP. Our main result is to show that Problem 1.1 is, in fact, NP complete, by establishing a reduction from the classical 3–SAT problem [2]. Specifically, given any Boolean expression involving n' variables, and in conjunctive normal form (product-of-sums), such that each clause contains three variables (possibly negated), we show how to construct, in time polynomial in n', an $n(n') \times n(n')$ output array for which (5) holds if and only if the given Boolean expression is satisfiable.

Our reduction from 3–SAT is based on implementing a Boolean expression as a circuit, involving "logic gates" and "wires," constructed from the $\{y_{i,j}\}$. The underlying $\{x_{i,j}\}$ can be interpreted as "signals" passing through the wires and gates. A subset of indices $V \subset \{(i,j)\}$ will correspond to the variables in the Boolean expression, and one index, (i_0, j_0), will correspond to the output of the Boolean expression. In our reduction, the $y_{i,j}$ for $(i,j) \notin V \cup \{(i_0, j_0)\}$ are chosen so that if $y_{i,j} = e$ for $(i,j) \in V \cup \{(i_0, j_0)\}$, the only input arrays $\{x_{i,j}\}$ for which the distance (5) evaluates to zero are precisely those for which $x_{i_0,j_0} = F(\{x_{i,j} : (i,j) \in V\})$, where F denotes the Boolean expression, and the Boolean 1 (true) and 0 (false) values are mapped, respectively, to $+1$ and -1. The reduction is completed by setting $y_{i_0+1,j_0} = 3$, which forces x_{i_0,j_0} to equal 1.

We note that the above MLSD problem can be cast as a maximum *a posteriori* probability estimation problem in a certain Bayesian belief network. This problem was shown to be NP complete in [3] for a variety of network constraints, all of which are less constrained than the specific network and joint distribution considered here. The present results are, therefore, not implied by [3]. Another related work is [4], which proves that multiuser detection, when the intra-user correlation matrix is unconstrained and is part of the problem instance, is NP complete. In contrast, we show NP completeness when only the channel outputs comprise the problem instance for a specific 3-tap 2D ISI. As shown in Section 5, our reduction for the 2D case can be used to establish NP completeness for a constrained version of multiuser detection (shown to be equivalent to a single-user detection problem with growing ISI) left open in [4].

In the next section, we describe the various components underlying our reduction, such as vertical and horizontal wires, various gates, and inverters. In Section 3, we describe how these components are combined to obtain the overall reduction. Then in Section 4, we show how to modify the reduction so that it applies to an additive white Gaussian noise (AWGN) channel instead of (2). In the distance metric (3) corresponding to this case, $d(y, u)$ is set to the squared error $(y - u)^2$. Finally, in Section 5 we show how the reduction presented in Section 4 can be used to prove Conjecture 1 in [4] stating that multiuser detection with a Toeplitz constraint is NP complete.

2 Reduction components

Figures 1 through 7 illustrate the building blocks of our reduction. The figures depict subarrays of inputs and outputs that can be replicated, extrapolated, and tiled to obtain larger input and output arrays. Most cells of the depicted subarrays contain a pair of values separated by a forward slash, such as $1/x$. The left and right values of the pair correspond respectively to channel output and input arrays. A cell labeled with $1/x$ for example, specifies that the corresponding channel output array location is set to 1 while the input location has the undetermined value $x \in \{-1, 1\}$. In the figures and the sequel we use standard notation from Boolean logic, such as a bar over a variable to denote inversion, and $\wedge$ and $\oplus$ to denote AND and XOR (exclusive OR), where $+1$ and -1, are interpreted as logical 1 and 0, respectively.

			$\vdots$	$\vdots$				
			$3/1$	$1/x$				
			$3/1$	$1/\overline{x}$				
			$3/1$	$1/x$				
			$3/1$	$1/\overline{x}$	$e/1$	$3/1$	$3/1$	$\ldots$
$\ldots$	$3/1$	$3/1$	$3/1$	$1/x$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$\ldots$
$\ldots$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$1/1$				
				$3/1$				

Figure 2: Wires: upward T–connection.

		e/y
	e/w	$1/y'$
e/x	$1/x'$	$1/z$

x	y	z	x'	y'	w
-1	-1	-1	1	1	1
-1	1	1	1	-1	1
1	-1	1	-1	1	1
1	1	-1	1	1	-1

Figure 3: Universal gate and truth table.

For each figure, the specified set of channel output values induces a constraint on input arrays for which the corresponding ISI output satisfies (5). The constrained input values for each figure are denoted by variables and the constraints are denoted using Boolean operations. Thus, in Figure 1 the input cells labeled x and $\overline{x}$ are constrained to be negations of one another if the resulting ISI is to achieve a distance metric of zero relative to the specified output values. The figures depict certain choices of output values that induce constraints which are useful for our Boolean reduction. Figures 1 and 2 represent "wires" which serve to constrain the input values of arbitrarily distant groups of cells. The figures illustrate how "wires" can bend and branch in a variety of directions. Figure 3 implements a variety of logic gates computing Boolean operations on the variables x and y. The truth table in the bottom half of the figure specifies the constraints imposed on the corresponding input variables

			$\vdots$	$\vdots$				
			$3/1$	$1/y$				
			$3/1$	$1/\overline{y}$				
			$3/1$	$1/y$	$e/1$	$3/1$	$3/1$	$\ldots$
$\ldots$	$3/1$	$3/1$	e/ϕ	$1/\overline{x \wedge y}$	$1/x \wedge y$	$1/\overline{x \wedge y}$	$1/x \wedge y$	$\ldots$
$\ldots$	$1/x$	$1/\overline{x}$	$1/\phi$	$1/\phi$				

Figure 4: AND gate with leads.

			$\vdots$	$\vdots$				
			$3/1$	$1/y$				
			$3/1$	$1/\overline{y}$				
			$3/1$	$1/y$				
$\ldots$	$3/1$	$3/1$	e/ϕ	$1/\phi$	$e/1$	$3/1$	$3/1$	$\ldots$
$\ldots$	$1/\overline{x}$	$1/x$	$1/\phi$	$1/x \oplus y$	$1/\overline{x \oplus y}$	$1/x \oplus y$	$1/\overline{x \oplus y}$	$\ldots$

Figure 5: XOR gate with leads.

appearing in the cells. The variable z, for example, is constrained to be $\overline{x \oplus y}$ while the variable y' is constrained to be $\overline{\overline{x} \wedge y}$. Figures 4 and 5 illustrate how “wires” can be attached to the input and output terminals of the “universal gate” of Figure 3 to implement wired binary AND and XOR gates, respectively. Finally, Figures 6 and 7 depict “inverters” that can be used to replace small portions of sufficiently long wires in the horizontal and vertical directions to invert the constraint between input variables at the two ends of the wire.

In all of the figures, an input value of ϕ corresponds to a value that is constrained by the specified output values, but which is not relevant to the reduction. An unspecified input value for a cell is unconstrained by the specified outputs, and can be assumed to be 1 (e.g. cell containing second e in third row of Figure 1). To avoid cluttering the figures we also avoid specifying input values that are

				3/1				
...	3/1	3/1	$e/\overline{x}$	$1/x$	$e/1$	3/1	3/1	...
...	$1/\overline{x}$	$1/x$	1/1	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$1/x$	...

Figure 6: Horizontal inverter.

		$\vdots$	$\vdots$	
		3/1	$1/y$	
		3/1	$1/\overline{y}$	
		3/1	$1/y$	
		$e/\overline{y}$	1/1	
	3/1	$1/y$	$1/\overline{y}$	
		$e/1$	$1/y$	
		3/1	$1/\overline{y}$	
		3/1	$1/y$	
		$\vdots$	$\vdots$	

Figure 7: Vertical inverter.

adjacent to the main structures and are obviously constrained to be 1 by neighboring outputs set to 3 (e.g. top row of empty cells in Figure 1). All unspecified output values are assumed to be e.

3 The full reduction

It can be shown that for any Boolean expression (in conjunctive normal form satisfying the 3–SAT restriction), an output array can be constructed from the basic components of the previous section, in time polynomial in the size of the Boolean expression, such that the corresponding set of constrained input array variables constitute a circuit (with "wires," "gates," and "inverters") implementing the Boolean expression. More precisely, the input variable corresponding to the circuit output is constrained to be the Boolean expression evaluated on the input variables corresponding to the circuit inputs, if

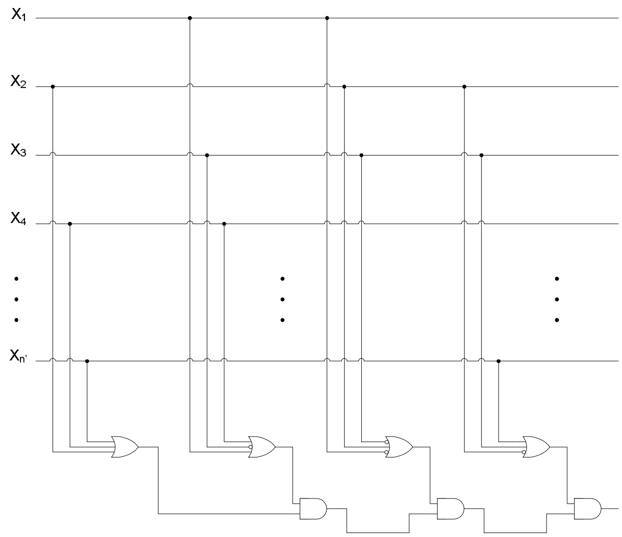

Figure 8: Schematic for full reduction of 3–SAT Boolean equation.

and only if the associated ISI is within a distance metric of zero from the specified output array. The reduction is completed by constraining the circuit output variable to be 1 by setting to 3 a channel output value whose filter input set contains the circuit output variable.

The basic architecture of the full reduction is depicted in Figure 8. It consists of a set of horizontal wires that propagate the circuit input values and a set of vertical wires that tap into these input values and collect them into sums along the bottom, the outputs of which, in turn, are collected into a product, one output at a time. The OR gate can be implemented using the 2–input AND gate construction of Figure 4 and inversion (Figures 6 and 7). We note that there are places in Figure 8 where horizontal and vertical wires cross. Such a crossing point can be implemented using the XOR gate of Figure 5, as illustrated in Figure 9.

4 Gaussian noise

In this section, we consider the MLSD problem for an additive white Gaussian noise channel with the same ISI of the previous section. The problem is essentially (3) again, but with real-valued channel outputs $\{y_{i,j}\}$, and $d(y, u)$ redefined from (4) to be the squared error distance

$$d(y, u) = (y - u)^2.$$

The reduction of 3–SAT from the previous sections does not apply immediately to this case, since

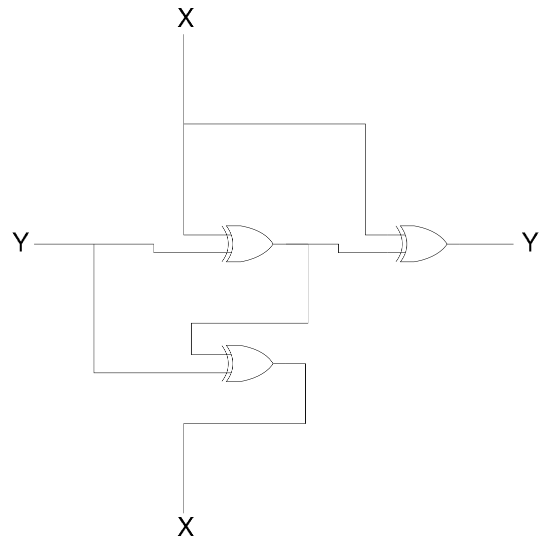

Figure 9: Crossing wires using XOR gates.

erasures (output symbols with 0 distance to all input symbols) are no longer available for building the channel output array. One approach to modifying the reduction is to replace erasures with channel outputs that behave partially like erasures, namely the set $\mathcal{E} \stackrel{\triangle}{=} \{-2, 0, 2\}$, which lie half way between ISI output values and thus have equal squared error distance to precisely two ISI output values. Since these output values do not have constant distance with the other two ISI output values, it is not immediate that such a modified reduction would go through.

We must examine the extent to which the reduction relies on the full equidistant property of erasures. There are several types of erasures, as distinguished by the nature of the constraints on the set of channel input values contributing to the corresponding ISI output value. We refer to this latter set as the ISI input set associated with the erasure. For $t = 0, 1, 2, 3$, a *type–t* erasure has an ISI input set with t values that are constrained, set, or assumed to be 1, in the above reduction. It can be checked that the reduction contains mostly type–3 erasures, a large number of type–2 erasures, and an isolated number of type–1 erasures. There are no type–0 erasures. Type–3 erasures can be found among the erasures that are used to fill in the "gaps" between wires and gates, and whose corresponding inputs are assumed to be 1. Examples of type–2 erasures are the ones implicit in the empty cells neighboring the x and $\overline{x}$ components of the wires in Figure 1. There are only two sources of type–1 erasures in the elementary reduction components. These are the second erasure from the left in the third row of Figure 1 and the implicit erasure to the right of the rightmost cell containing $1/\phi$ in the second row

						$3/1$				
$\dots$	$3/1$	$3/1$	$3/1$	$3/1$	e/x	$1/\overline{x}$	$e/1$	$3/1$	$3/1$	$\dots$
$\dots$	$1/x$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$1/1$	$1/x$	$1/\overline{x}$	$1/x$	$1/\overline{x}$	$\dots$
			$e/1$	$1/x$	$e/$					
			$3/1$	$1/\overline{x}$						
			$3/1$	$1/x$						
			$3/1$	$1/\overline{x}$						
			$3/1$	$1/x$						
			$\vdots$	$\vdots$						

Figure 10: Modified downward T–connection.

			$\vdots$	$\vdots$								
			$3/1$	$1/y$								
			$3/1$	$1/\overline{y}$				$3/1$				
			$3/1$	$1/y$	$e/1$	$3/1$	$e/x\wedge y$	$1/\overline{x\wedge y}$	$e/1$	$3/1$	$3/1$	$\dots$
$\dots$	$3/1$	$3/1$	e/ϕ	$1/\overline{x\wedge y}$	$1/x\wedge y$	$1/\overline{x\wedge y}$	$1/1$	$1/x\wedge y$	$1/\overline{x\wedge y}$	$1/x\wedge y$	$1/\overline{x\wedge y}$	$\dots$
$\dots$	$1/x$	$1/\overline{x}$	$1/\phi$	$1/\phi$	$1/\phi$	$1/\phi$	$e/$					

Figure 11: Modified AND gate with leads.

from the bottom of Figure 4.

A Type–3 erasure can clearly be changed to an output value of 3, while fully preserving the reduction. Additionally, it is clear that there exists an input array satisfying (5) with the reduction induced outputs if and only if there is one in which the ISI output corresponding to a type–2 erasure is either 3 or 1. This implies that the reduction goes through if we replace a type–2 erasure by 2 and increase the value against which the distance is being compared in Problem 1.1 by 1. Type–1 erasures, on the other hand, are more problematic. Consider the first type–1 erasure mentioned above. It is not hard to see that if there exists an input array satisfying (5) then there is one in which the corresponding ISI output can be one of -1 and 3, or one of -3 and 1. This follows from the fact that the two diagonally related ISI inputs of this erasure are constrained to be equal. The channel output values which are equidistant to the above pairs of ISI outputs are 1 and -1, respectively, and either would contribute the same distance with respect to any input array satisfying (5). Both potential output values are problematic, however, since a distance contribution of zero could be achieved at this location by violating the equality constraint on the diagonally related ISI inputs for the erasure. This raises the possibility of the existence of an input array meeting the increased minimum distance threshold even in the case that the Boolean expression is not satisfiable. The other type–1 erasure mentioned above causes similar difficulties.

Therefore, it seems that type–1 erasures cannot, in general (and in the above specific cases) be replaced by real output values that preserve the reduction. We get around this difficulty by modifying the reduction components of Figures 1 and 4 to eliminate type–1 erasures. The somewhat more complicated versions are displayed in Figures 10 and 11. It can be checked that all erasures appearing in these modified components are type–2 or type–3, and further, that the specified outputs do indeed impose the displayed constraints on channel inputs. It is also evident that the modified components can be used to implement the same functionality in the reduction as their original counterparts.

Thus, the overall reduction for the squared error case follows the reduction of the previous case, with the following changes:

1. The modified components in Figures 10 and 11 replace their counterparts in Figures 1 and 4.

2. Type–3 erasures are replaced by 3's.

3. Type–2 erasures are replaced by 2's.

4. Increase the distance threshold in the (decision) Problem 1.1 by the number of type–2 erasures.

It should be clear that the modified reduction can be implemented in polynomial time, and that it indeed decides the 3–SAT problem.

5 Multiuser detection and 1D MLSD

In [4] the (binary input) synchronous multiuser detection problem is reduced to

Problem 5.1

$$\arg \max_{\mathbf{b}\in\{+1,-1\}^k} 2\mathbf{b}^t\mathbf{z} - \mathbf{b}^t\mathbf{H}\mathbf{b}, \tag{6}$$

where $\mathbf{H}$ is a $k \times k$ nonnegative definite matrix and $\mathbf{z}$ is a $k \times 1$ vector. It is shown in [4] that this problem is NP hard when a problem instance consists of $\mathbf{z}$ and $\mathbf{H}$ with rational entries.

It is further shown that the single-user intersymbol interference problem corresponding to determining the maximum *a posteriori* (MAP) estimate of $b(-M), b(-M+1), \ldots, b(M) \in \{+1, -1\}$ upon observing

$$r(t) = \sum_{i=-M}^{M} b(i)s(t - iT) + \nu(t)$$

for $t \in (-\infty, \infty)$, with $s(t)$ having finite support, $\nu(t)$ being white Gaussian noise, and $\{b(i)\}$ i.i.d. uniform Bernoulli $\{+1, -1\}$, is a special case of the multiuser detection problem with the restriction that $\mathbf{H}$ is Toeplitz. Conjecture 1 on p. 311 in [4] states that (6) remains NP hard even when $\mathbf{H}$ is restricted to be Toeplitz, implying the intractability of the single-user intersymbol interference problem with a growing number of interfering signals (the case of a bounded number of interfering signals is readily handled by the Viterbi algorithm). We now show that a slight variation of the reduction of the previous section can also be used to reduce Boolean satisfiability to a decision version of (6) with $\mathbf{H}$ Toeplitz, thereby proving Conjecture 1 of [4].

Given an $n \times n$ array $\mathbf{x} = \{x_{i,j}\}$, let $v(\mathbf{x})$ denote the $n^2 \times 1$ vector whose m-th entry is $x_{i(m),j(m)}$, where $i(m) = 1 + \lfloor (m-1)/n \rfloor$ and $j = m - n\lfloor (m-1)/n \rfloor$. The vector $v(\mathbf{x})$ thus consists of a concatenation of the transposed rows of $\mathbf{x}$. Let $\tilde{h}_i = 1$ for $i = 1$, 2, and $n+1$, and $\tilde{h}_i = 0$ for all other $i \le n$. Let $\tilde{\mathbf{H}}$ be an $(n^2 + n) \times n^2$ lower-triangular Toeplitz matrix with $\tilde{h}_i$ along the $(i-1)$-st lower diagonal, $i = 1, 2, \ldots, n+1$, and 0's along the remaining diagonals:

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{h}_1 & 0 & & 0 & 0 \\ \tilde{h}_2 & \tilde{h}_1 & & \vdots & 0 \\ \vdots & \tilde{h}_2 & \ddots & 0 & \vdots \\ \tilde{h}_{n+1} & \vdots & & \tilde{h}_1 & 0 \\ 0 & \tilde{h}_{n+1} & & \tilde{h}_2 & \tilde{h}_1 \\ 0 & 0 & \ddots & \vdots & \tilde{h}_2 \\ \vdots & \vdots & & \tilde{h}_{n+1} & \vdots \\ 0 & 0 & & 0 & \tilde{h}_{n+1} \end{bmatrix}.$$

For $(i,j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$, let $u_{i,j} = x_{i,j} + x_{i-1,j} + x_{i,j-1}$ be the array $\mathbf{x}$ corrupted by the 2D ISI of the previous sections, where $x_{i,j}$ is assumed to be 1 when either $i = 0$ or $j = 0$. It can be checked that the corresponding components of $v(\mathbf{u})$ and $\tilde{\mathbf{H}}v(\mathbf{x})$ are equal for the set of indices

$$I = \{(i-1)n + j : 1 < i \leq n, 1 < j \leq n\}.$$

For the remaining indices, which correspond to borders with either $i = 1$ or $j = 1$ or the extra n indices of $\tilde{\mathbf{H}}v(\mathbf{x})$, the components of $\tilde{\mathbf{H}}v(\mathbf{x})$ are linear combinations of one or more boundary components of $\mathbf{x}$ (indices with $j = 1$, $j = n$, $i = 1$, or $i = n$).

Given an instance of 3–SAT of size n', it is evident that an $n \times n$ output array $\mathbf{y}$ can be constructed according to the polynomial-time reduction in Section 4 in which the boundary components of the array are not part of the "circuit" implementing the reduction, and are set to the value 3. Let $\tilde{\mathbf{y}}$ be an $(n^2 + n) \times 1$ vector which agrees with $v(\mathbf{y})$ in all components with indices in I and is equal to $\tilde{\mathbf{H}}\mathbf{1}$ in the remaining components, where $\mathbf{1}$ is a vector of all 1's. It can be checked that the resulting $\tilde{\mathbf{y}}$ has the property that

$$\min_{\mathbf{x}} \sum_{1 \leq i,j \leq n} d(y_{i,j}, x_{i,j} + x_{i-1,j} + x_{i,j-1}) = T \tag{7}$$

with T set to the number of 2's in $\mathbf{y}$ (and $d(\cdot,\cdot)$ being the squared error distance), if and only if

$$\min_{\tilde{\mathbf{x}}} ||\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}||_2^2 = T \tag{8}$$

where $||\cdot||_2$ denotes the L_2 norm on vectors, and the minimizations in (7) and (8) are, respectively, over $n \times n$ and $n^2 \times 1$ arrays of $\{+1, -1\}$. Note that any vector $\tilde{\mathbf{x}}$ achieving the minimum in (8) would necessarily have 1's in all components effecting those components of $\tilde{\mathbf{H}}\tilde{\mathbf{x}}$ not in I.

The reduction to a decision version of (6) is completed by noting that

$$||\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{x}}||_2^2 = \tilde{\mathbf{y}}^t\tilde{\mathbf{y}} - 2\tilde{\mathbf{x}}^t\tilde{\mathbf{H}}^t\tilde{\mathbf{y}} + \tilde{\mathbf{x}}^t\tilde{\mathbf{H}}^t\tilde{\mathbf{H}}\tilde{\mathbf{x}}$$

so that (8) is equivalent to

$$\max_{\mathbf{b} \in \{+1,-1\}^{n^2}} 2b^t\mathbf{z} - \mathbf{b}^t\mathbf{H}\mathbf{b} = T' \tag{9}$$

where $\mathbf{H} = \tilde{\mathbf{H}}^t\tilde{\mathbf{H}}$, $\mathbf{z} = \tilde{\mathbf{H}}^t\tilde{\mathbf{y}}$, and $T' = \tilde{\mathbf{y}}^t\tilde{\mathbf{y}} - T$. It can be checked that $\tilde{\mathbf{H}}^t\tilde{\mathbf{H}}$ is Toeplitz and non-negative definite. This completes the (polynomial-time) reduction of 3–SAT to an instance of a decision version of (6) (which asks if (9) holds) with the added constraint that $\mathbf{H}$ is Toeplitz.

Acknowledgments

We gratefully acknowledge Sergio Verdú for helpful discussions about [4].

References

[1] K. M. Chugg, "Performance of optimal digital page detection in a two-dimensional ISI/AWGN channel," *Proceedings of the 30th Asilomar Conference on Signals, Systems, and Computers*, pp. 958–962, 1996.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP Completeness*, W. H. Freeman and Co. 1979.

[3] S. E. Shimonyi, "Finding MAPs for belief networks is NP hard," *Artificial Intelligence*, Vol. 68, No. 2, pp. 399–410, August, 1994.

[4] S. Verdú, "Computational complexity of optimum multiuser detection," *Algorithmica*, 4: 303–312.