The Parameterized Complexity of Some Fundamental Problems of Linear Codes and Integer Lattices

Michael R. Fellows
Department of Computer Science
University of Victoria
Victoria, B.C. V8W 3P6 Canada
mfellows@csr.uvic.ca

Rod Downey and *Geoff Whittle*
Department of Mathematics
Victoria University
Private Bag 600
Wellington, New Zealand
{downey,geoff.whittle}@math.vuw.ac.nz

Alexander Vardy
Coordinated Science Laboratory
University of Illinois
1308 W. Main St., Urbana, IL 61801 U.S.A.
vardy@golay.csl.uiuc.edu

March 19, 1997

Abstract

The parameterized complexity of a number of fundamental problems of linear codes and integer lattices is explored. Concerning codes, the main results are that MAXIMUM LIKELIHOOD DECODING and WEIGHT DISTRIBUTION are hard for $W[1]$ and NP by parametric polynomial-time transformations from the PERFECT CODE problem in graphs. Concerning lattices, we prove a similar result for the THETA SERIES problem of determining, for an integer lattice L, given by a set of generators, and a positive integer k, whether there is a vector x in L of Euclidean norm k. This problem is closely related to the SHORTEST VECTOR problem that has assumed great importance due to the results of Ajtai on the generation of hard instances. From the $W[1]$-hardness of MAXIMUM LIKELIHOOD DECODING, WEIGHT DISTRIBUTION and THETA SERIES, it is concluded by a theorem of Cai and Chen that these problems do not have fully polynomial-time approximation schemes unless the parameterized complexity hierarchy collapses.

1 Introduction

Our basic purpose in this paper is to explore the parameterized complexity of fundamental computational problems in the theories of linear codes and integer lattices. There is a natural close relationship between computational problems in these areas. We prove one main combinatorial transformation which we then use to show hardness for problems in both areas.

There has been a substantial amount of previous work on the complexity of the problems considered here in the framework of *NP*-completeness. Many of these problems are naturally parameterized. The following basic problems of linear codes were considered by Berlekamp, McEliece and van Tilborg:

Maximum Likelihood Decoding
Instance: A set H of binary vectors of length m, a target vector s of length m, and a positive integer k.
Parameter: k
Question: Is there a set of at most k vectors of H that sum to s?

Weight Distribution
Instance: A set H of binary vectors of length m, and a positive integer k.
Parameter: k
Question: Is there a set of k vectors of H that sum to the all-zero vector?

Minimum Distance
Instance: A set H of binary vectors of length m, and a positive integer k.
Parameter: k
Question: Is there a non-empty set of at most k vectors of H that sum to the all-zero vector?

In 1978, Berlekamp, McEliece and van Tilborg [BMcEvT78] proved that Maximum Likelihood Decoding and Weight Distribution are *NP*-complete, by a reduction from 3-Dimensional Matching. The Minimum Distance problem has recently been shown to be *NP*-complete by Vardy [Var97], also by a reduction from 3-Dimensional Matching. Because 3-Dimensional Matching is fixed-parameter tractable, these earlier results not allow us to conclude anything about the parameterized complexity of the above problems.

Note that the difference between the definitions of the Weight Distribution and Minimum Distance problems is very slight. Weight Distribution requires *exactly* k vectors and Minimum Distance requires *at least* k vectors in a solution.

We prove that Maximum Likelihood Decoding and Weight Distribution are hard for $W[1]$ by a parametric polynomial-time reduction from the Perfect Code problem for graphs. (Such a reduction proves both $W[1]$-hardness and *NP*-completeness at the same time.) Both problems are also shown to belong to $W[2]$.

There are two analogous problems in the theory of integer lattices, defined as follows.

Theta Series
Instance: A basis $X = \{x_1, \ldots, x_n\} \subseteq \mathbf{Z}^n$ for a lattice L, and a positive integer k.
Parameter: k
Question: Is there a non-zero vector $x \in L$, such that $||x||^2 = k$?

Shortest Vector
Instance: A basis $X = \{x_1, \ldots, x_n\} \subseteq \mathbf{Z}^n$ for a lattice L, and a positive integer k.
Parameter: k

Question: Is there a non-zero vector $x \in L$, such that $||x||^2 \leq k$?

We prove here for the first time that THETA SERIES is *NP*-hard. Our transformation also shows that it is hard for $W[1]$.

P. van Emde Boas conjectured in 1980 [vEB80] that SHORTEST VECTOR is *NP*-hard. Despite considerable effort (see [ABSS93] and [Var97] for a discussion), this conjecture has been — and remains — open. Our result on THETA SERIES adds weight to the conjecture, and we believe that the methods we employ here have some promise in addressing this important problem.

The significance of the SHORTEST VECTOR problem has been dramatically increased by the results of Ajtai, who showed in [Aj96] that integer lattices can be randomly generated efficiently in such a way that if there is a *P*-time probabilistic algorithm that solves the SHORTEST VECTOR problem on the generated instance distribution with probability 1/2, then there is a *P*-time probabilistic algorithm that solves SHORTEST VECTOR for every lattice, with probability exponentially close to 1. Algorithms for approximately solving the SHORTEST VECTOR problem have many uses (see [Lov86] and [Kan87] for surveys).

Our results are based on a combinatorial transformation that has many similarities to the proofs of the main theorems in [DF95a] and [DF95b]. We feel that one of the most interesting aspects of our work is a demonstration of the potential utility of parametric methods and perspectives in addressing issues in "classical" complexity. We will assume that the reader has a basic familiarity with the parameterized complexity framework, such as can be found in [DF95a, DF95b, DF95c, DF97]. Readers who do not have this background can find some discussion and essential definitions in Appendix 1.

In the interests of readability we defer the proof of our main combinatorial transformation (Theorem 1) to §3. In the next section we prove the main results, using Theorem 1. In §4 we prove membership in $W[2]$ for several of the problems considered. In the concluding section we discuss remaining open problems, and speculate on whether the techniques used here might be adapted to provide a proof of $W[1]$-hardness for MINIMUM DISTANCE, or *NP*-hardness for SHORTEST VECTOR.

2 The Main Results Via Red/Blue Graph Problems

The starting point for our transformations is the parameterized PERFECT CODE problem. This was shown to be hard for $W[1]$ in [DF95b]. It was first proved *NP*-complete by Kratochvíl [KK88, Kra94].

We will use the following graph-theoretic notation and terminology. If $G = (V, E)$ is a graph and $u \in V$ is a vertex of G, then the *open neighborhood* $N(u)$ of u is defined to be $N(u) = \{v : (u, v) \in E\}$. The *closed neighborhood* $N[u]$ of u is $N[u] = N(u) \cup \{u\}$. We say that two vertices in a graph are *close* if a shortest path between them has length 1 or 2.

PERFECT CODE

Instance: A graph $G = (V, E)$ and a positive integer k.
Parameter: k
Question: Is there a k-element *perfect code* in G? (That is, a set of k vertices $V' \subseteq V$ such that for every vertex $u \in V$, there is a unique $v \in V'$ with $u \in N[v]$.)

Our main tool is a way of describing and manipulating codes and other sets of vectors by means of bipartite graphs. Suppose $G = (R, B, E)$ is bipartite graph with the partition of the vertices into the red set R and the blue set B. We make the following definitions concerning special sets of vertices in G.

Definition. Suppose $G = (R, B, E)$ is a red/blue bipartite graph and $R' \subseteq R$ is a nonempty set of red vertices.
R' is a *dominating set* if

$$\forall u \in B : |R' \cap N(u)| > 0$$

R' is an *odd set* if

$$\forall u \in B : |R' \cap N(u)| \text{ is odd}$$

R' is a *perfect code* if

$$\forall u \in B : |R' \cap N(u)| = 1$$

R' is an *even set* if

$$\forall u \in B : |R' \cap N(u)| \text{ is odd}$$

R' is a **Z**-*even* set if

$$\exists c : R' \to \mathbf{Z} \backslash \{0\} \quad \forall u \in B : \sum_{v \in N(u) \cap R'} c(v) = 0$$

Note that in a red/blue bipartite graph, what we define to be a dominating set is not the same thing as a dominating set in the underlying uncolored graph.

We can now state the main problems about red/blue graphs that we will deal with.

Exact Even Set
Instance: A red/blue graph $G = (R, B, E)$ and a positive integer k.
Parameter: k
Question: Is there a set of k vertices $R' \subseteq R$ that is an even set in G?

Even Set
Instance: A red/blue graph $G = (R, B, E)$ and a positive integer k.
Parameter: k
Question: Is there a set of at most k vertices $R' \subseteq R$ that is an even set in G.

Similarly, we define the problems: Odd Set, Exact Odd Set, **Z**-Even Set, and Exact **Z**-Even Set.

In the next section we prove the following theorem by means of a (somewhat complicated) combinatorial transformation that has a general architecture very similar to the main theorems of [DF95a, DF95b].

Theorem 1. Let $G =$ be a graph on n vertices, and k a positive integer. In time polynomial in n and k we can produce a red-blue bipartite graph G', and a positive integer k' such that:
(1) G has a perfect code of size k if and only if G' has a perfect code of size k'.
(2) Every dominating set of G' has size at least k'.
(3) Every dominating set of G' of size k' is a perfect code in G'.

This can be applied almost immediately to yield significant results that illustrate our graph-theoretic approach to problems of codes and lattices. First of all, note that we have the following merely by observing that in a red/blue graph a perfect code is an odd set, and an odd set is necessarily a dominating set.

Theorem 2. There is a polynomial-time parameterized transformation from PERFECT CODE to EXACT ODD SET and to ODD SET. □

In describing vectors and sets of vectors we will use the following notation. If $j \in \mathbf{Z}$, then j^r denotes the vector of length r having the value j in each component. For a vector x, we will write $x[i]$ to denote the ith component of x. If x is a vector of length n_1 and y is a vector of length n_2, then $x \cdot y$ denotes the vector of length $n_1 + n_2$ formed by concatenating x and y.

If $G = (R, B, E)$ is a red/blue graph, $u \in R$, and $\ell \in \mathbf{Z}$, then we define the length $|B|$ vector $x_u = x_u^\ell$:

$$x_u^\ell[v] = \ell \text{ if and only if } v \in N(u), \text{ and } x_u^\ell[v] = 0 \text{ otherwise}$$

In this situation, define

$$\mathcal{V}_G^\ell = \{x_u^\ell : u \in R\}$$

In other words, $\mathcal{V}_G^\ell$ consists of the rows of the red/blue adjacency matrix of G with adjacencies indicated by the entry ℓ.

Theorem 3. MAXIMUM LIKELIHOOD DECODING is hard for $W[1]$.

Proof. Using Theorem 1, we transform from the $W[1]$ and NP-hard PERFECT CODE problem. Suppose (G, k) is transformed into (G', k'). The image instance of MAXIMUM LIKELIHOOD DECODING consists of the set of vectors $\mathcal{V} = \mathcal{V}_{G'}^1$, the parameter $w = k'$ and the target vector $s = \mathbf{1}^{|B|}$.

If G has a k-element perfect code, then by (1) of Theorem 1, G' has a k'-element perfect code (in the red/blue sense), $R' \subseteq R$. The sum (in the field $GF[2]$) of the vectors x_u over $u \in R'$ is $\mathbf{1}^{|B|}$.

Conversely, suppose S is a set of k' vectors in $\mathcal{V}$ that sums to $\mathbf{1}^{|B|}$ over $GF[2]$. It is obvious that the corresponding set R_S of red vertices must be a dominating set in G'. By (3) of Theorem 1, R_S is a perfect code in G', and by (1), there is a k-element perfect code in G. □

Theorem 4. WEIGHT DISTRIBUTION is hard for $W[1]$.

Proof. We reduce from PERFECT CODE using the transformation of Theorem 1. Suppose (G, k) is transformed into $(G' = (R, B, E), k')$. Identify R with $\{1, ..., |R|\}$. For $u \in R$, let $\delta^1(u)$ denote the vector of length $|R|$ vector that is 1 in the uth component and otherwise 0. We reduce to the

instance of WEIGHT DISTRIBUTION described:
(1) The set of vectors (of length $2|B|+|R|$) is

$$\mathcal{V} = \mathcal{V}_1 \ \cup \ \mathcal{V}_2$$

where

$$\begin{aligned} \mathcal{V}_1 &= \{x_u^1 \cdot \delta^1(u) \cdot \mathbf{0}^{|B|} : \ u \in R\} \\ \mathcal{V}_2 &= \{\mathbf{0}^{|B|} \cdot \delta^1(u) \cdot x_u^1 : \ u \in R\} \\ z &= \mathbf{1}^{|B|} \cdot \mathbf{0}^{|R|} \cdot \mathbf{1}^{|B|}\} \end{aligned}$$

(2) The parameter is $w = 2k'+1$.

If G has a k-element perfect code, then by (1) of Theorem 1, G' has a k'-element perfect code R'. It is straightforward to verify that the $2k'$ vectors corresponding to R' in $\mathcal{V}_1$ and $\mathcal{V}_2$, together with the vector z sum to the all-zero vector over $GF[2]$.

In the other direction, suppose S is set of w vectors of $\mathcal{V}$ that sum to the all-zero vector over $GF[2]$. It is obvious that if S contains a vector x of $\mathcal{V}_1$, then S must also contain the corresponding vector x' of $\mathcal{V}_2$ else there is no way for the sum to be 1 in the middle component that is 1 only in x and x'. Thus the vertices of $S \cap (\mathcal{V}_1 \cup \mathcal{V}_2)$ are paired, and we must have $z \in S$.

This implies that $S \cap \mathcal{V}_1$ corresponds to a set of k' vertices V_S in G' that is a k'-element dominating set in G'. By (3) of Theorem 1, V_S is a perfect code in G', and by (1) of Theorem 1, G has a perfect code of size k. □

Theorem 5. EXACT EVEN SET is hard for $W[1]$ and NP-complete.

Proof. This follows from the proof of Theorem 4 by simply viewing the set of vectors $\mathcal{V}$ as the rows of an adjacency matrix for a red/blue graph. □

By modifying the argument for Theorem 4 only slightly (and interpreting things in terms of red/blue graphs), we have the following result.

Theorem 6. EXACT **Z**-EVEN SET is hard for $W[1]$ and NP.

Proof. We first describe a set of 0-1 vectors that is much the same as in the proof of Theorem 4, except that the length of the vectors is increased by 2 (the first two components in the description below), and there are 3 "special" vectors instead of one.

Suppose (G,k) is transformed to $(G' = (R,B,E),k')$ by the transformation of Theorem 1. Let $\delta^1(u)$ be the same as in the proof of Theorem 4. We transform to the instance of (G'',w) of EXACT **Z**-EVEN SET, where G'' is the red/blue graph obtained by interpreting the following set $\mathcal{V}$ of 0-1 vectors of length $2|B|+|R|+2$ as the red/blue adjacency matrix of G'', and where the parameter is $w = 2k'+3$:

$$\mathcal{V} = \mathcal{V}_1 \ \cup \ \mathcal{V}_2 \ \cup \mathcal{V}_3$$

where

$$\mathcal{V}_1 \ = \ \{(0,0) \cdot x_u^1 \cdot \delta^1(u) \cdot \mathbf{0}^{|B|} : u \in R\}$$

$$
\begin{aligned}
\mathcal{V}_2 &= \{(0,0)\cdot \mathbf{0}^{|B|}\cdot \delta^1(u)\cdot x_u^1 : u\in R\}\\
\mathcal{V}_3 &= \{z_1, z_2, z_3\}\\
z_1 &= (1,1)\cdot \mathbf{1}^{|B|}\cdot \mathbf{0}^{|R|+|B|}\\
z_2 &= (0,1)\cdot \mathbf{0}^{|B|+|R|}\cdot \mathbf{1}^{|B|}\\
z_3 &= (1,0)\cdot \mathbf{0}^{|B|+|R|+|B|}\}
\end{aligned}
$$

If G has a k-element perfect code, then by (1) of Theorem 1, G' has a k'-element perfect code R'. It is easy to verify that the following set of vectors of $\mathcal{V}$ (equivalently, vertices of G'') sum to the all-zero vector when multiplied first by the indicated coefficients (equivalently, are the vertices are a w-element $\mathbf{Z}$-even set in G'' as shown by the indicated weighting function):

- z_1 with the coefficient -1,
- z_2 and z_3, each with the coefficient $+1$,
- the k' vectors of $\mathcal{V}_1$ corresponding to R', each with the coefficient $+1$, and
- the k' vectors of $\mathcal{V}_2$ corresponding to R', each with the coefficient -1 .

We now consider the other direction. Suppose that S is a $\mathbf{Z}$-even set of size w in G'', and equivalently view S as a set of vectors of $\mathcal{V}$ for which there is a set of non-zero integer coefficients for the elements of S so that the sum is the all-zero vector. By the same argument as in the proof of Theorem 4, noting those components in which only two vectors in $\mathcal{V}$ have a non-zero entry, we see that each vector x of $\mathcal{V}_1$ is paired with a unique vector $x' \in \mathcal{V}_2$ in the sense that if either belongs to S then both do. The reason for the pairing is that if only one of a pair belonged to S, then in any sum with nonzero coefficients, the component that creates the pairing could not be zero. Furthermore note that the two coefficients of a pair in S must be opposites c and $-c$.

The vectors z_1 and z_2 are paired in the second component. Since w is odd, z_3 must belong to S. But also z_1 and z_3 are paired in the first component, and consequently all three of the z_i must belong to S.

There are then two cases, which come to the same thing by symmetry. Without loss of generality, we can thus suppose that the coefficient of z_1 is $-c$, the coefficient of z_2 is $+c$ and the coefficient of z_3 is $+c$ (in order to get the first two components to cancel out in the sum).

The k' vectors in $\mathcal{V}_1 \cap S$ are forced to represent a dominating set in G' in order for the components of the sum of index 3 through $|B|+2$ to be 0. By (3) and (1) of Theorem 1 we may then conclude that G has a k-element perfect code. $\square$

Theorem 7. THETA SERIES is hard for $W[1]$ and NP.

Proof. Let (G,k) be an instance of PERFECT CODE, let (G',k') be the result of the transformation of Theorem 1 and let $(G''=(R,B,E),w)$ be the instance of EXACT $\mathbf{Z}$-EVEN SET to which (G',k') is further transformed by the proof of Theorem 6. Identify R with $\{1,\ldots,|R|\}$. Let $\delta^2(u)$ denote the vector of length $|R|$ that has the value 2 in the uth component, and is otherwise 0. Choose $\ell > 4w$. Let $\delta^{\ell^2}(j)$ denote the vector of length $|B|$ that is ℓ^2 in the jth component and otherwise 0. The instance (L,w') of THETA SERIES to which we transform (G,k) is described:

(1) The basis of the lattice L is given by the set of vectors $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3$ where

$$\begin{aligned}
\mathcal{V}_1 &= \left\{x_u^{\ell} \cdot \delta^2(u) : u \in R \setminus \{z_1, z_2, z_3\}\right\} \\
\mathcal{V}_2 &= \{x_{z_i}^{\ell} \cdot \delta^1(z_i) : i = 1, 2, 3\} \\
\mathcal{V}_3 &= \{\delta^{\ell^2}(j) \cdot \mathbf{0}^{|R|} : 1 \leq j \leq |B|\}
\end{aligned}$$

(2) The parameter is $w' = 8k' + 3$.

The basis for L described above consists of $|R| + |B|$ vectors of length $|R| + |B|$ that are clearly a linearly independent set over $\mathbf{R}$.

If G has a k-element perfect code, then by (1) of Theorem 1, the proof of Theorem 6 and definition of $\mathcal{V}$, it is easy to see that there is a set of $2k' + 3$ vectors of $\mathcal{V}$ (inherited from the proof of Theorem 6) that form a linear combination x with coefficients taken from $\{+1, -1\}$ so that all of the first $|B|$ components of x are 0, $2k'$ of the last $|R|$ components are 2, and three of the last $|R|$ components (corresponding to z_1, z_2, z_3) are 1. Consequently $||x||^2 = 8k' + 3$.

In the converse direction, suppose S is a set of vectors of $\mathcal{V}$ such that there is a $\mathbf{Z}$-linear combination x of the elements of S (with non-zero coefficients) such that $||x||^2 = w'$.

Claim 1. None of the vectors of $\mathcal{V}_3$ belongs to S.

Proof. The first $|B|$ components of x are necessarily multiples of ℓ. Since $\ell^2 > w'$, it must be the case that x is 0 in these components. If S were to contain a vector in $\mathcal{V}_3$, then the absolute value of the sum of the coefficients multiplying the elements of S that belong to $\mathcal{V}_1 \cup \mathcal{V}_2$ must be at least ℓ. But this implies that $||x||^2 \geq \ell^2 > w'$. □

Claim 1 immediately forces the three vectors corresponding to z_1, z_2, z_3 to belong to S since otherwise $||x||^2$ cannot be congruent to 3 mod 4. Furthermore, as in the proof of Theorem 4, this situation enforces a pairing between the vectors in the two halves of $\mathcal{V}_1$ (see the proof of Theorem 4). The subset $S' \subseteq S$ in the first half of $\mathcal{V}_1$ (i.e., those vectors whose first $|B|$ components represent vertex neighborhoods in G') must represent a dominating set in G'. Since S' must contain at least k' elements (by (2) of Theorem 1), and is paired with a subset of S in the second half of $\mathcal{V}_1$ of the same size, S' must contain exactly k' elements. By (3) of Theorem 1 it therefore represents a perfect code in G', and so by (1) of Theorem 1 there is a k-element perfect code in G. □

A theorem of Cai and Chen [CC93] states that if the optimization problem naturally associated to an integer-parameter problem has a fully polynomial-time approximation scheme (see [GJ79] for an exposition of this concept), then the corresponding parameterized problem is fixed parameter tractable. We can draw the following corollary.

Corollary. There is no fully polynomial-time approximation scheme for any of the problems discussed in this section unless $FPT = W[1]$.

In the next section we prove our principal combinatorial transformations, Theorem 1.

3 A Transformation of the PERFECT CODE Problem

Theorem 1. Let $G = (V, E)$ be a graph on n vertices, and k a positive integer. In time polynomial in n and k we can produce a red-blue bipartite graph $G' = (R, B, F)$, and the positive integer

$$k' = (2k+1) + 3\left(2k(k+1) + (k+1)^2\right)$$

such that:
(1) G has a perfect code of size k if and only if G' has a perfect code of size k'.
(2) Every dominating set of G' has size at least k'.
(3) Every dominating set of G' of size k' is a perfect code in G'.

Our proof of Theorem 1 has many similarities with the proofs of the main theorems of [DF95a, DF95b] to which the reader may wish to refer.

Proof. In describing the construction of G' from G it is convenient to identify V with the set of integers $\{1, \ldots, n\}$. An *interval* of vertices is defined to be a subset consisting of consecutive elements, for example, $\{3, 4, 5, 6\}$, and may be empty. Let $\mathcal{J}$ denote the set of all nonempty intervals having a size of at most $n - k$:

$$\mathcal{J} = \{J : 1 \leq |J| \leq n - k,\ J \text{ is an interval of } V\}$$

For an interval $J \in \mathcal{J}$ define the *initial boundary* $\partial(J)$ of J to be the largest non-negative integer strictly less than the smallest element of J. Thus for example, $\partial(\{2, 3, 4\}) = 1$, $\partial(\{6\}) = 5$, $\partial(\{1, 2\}) = 0$. Define the *terminal boundary* $\partial'(J)$ to be the smallest positive integer strictly greater than the largest element of J. Thus, $\partial'(\{2, 3, 4\}) = 5$.

We will also need to refer to empty intervals, but we will still need to refer to where they begin and end. For this purpose we will consider the special symbols $\epsilon_0, \ldots, \epsilon_n$ and extend the definitions of ∂ and ∂' as follows: for $u = 0, \ldots, n$, $\partial(\epsilon_u) = u$ and $\partial'(\epsilon_u) = u + 1$. Thus ϵ_u represents the "interval" of vertices in V that is empty, but "located" between u and $u + 1$.

Let $\mathcal{J}'$ denote the set of nonempty intervals $\mathcal{J}$ augmented with these empty intervals:

$$\mathcal{J}' = \mathcal{J} \cup \{\epsilon_u : 0 \leq u \leq n\}$$

If $u, v \in V$ with $u < v$ then we define the interval *between* u and v to be the set $J(u, v) = \{u + 1, \ldots, v - 1\}$ if $v - u \geq 2$, and if $v = u + 1$ then $J(u, v) = \epsilon_u$. Similarly, for $u \in V$ define the interval *preceding* u to be $J(0, u) = \{1, \ldots, u - 1\}$ if $u \geq 2$, and if $u = 1$ then $J(0, u) = J(0, 1) = \epsilon_0$. The interval *succeeding* $u \in V$ is defined: $J(u, \infty) = \{u + 1, \ldots, n\}$ if $u < n$, and $J(n, \infty) = \epsilon_n$.

Let $[k]$ denote the set of integers $\{1, \ldots, k\}$ and let $[k]^*$ denote the set of integers $\{0, \ldots, k\}$. Parts of our construction of G' will be quantified over the set of ordered pairs of elements of $[k]^*$. To have an easy way of refering to these, let

$$\Delta = \{(i, i') : i, i' \in [k]^*\}$$

Fix a circular ordering of Δ, such as obtained by wrapping the lexicographic ordering. We will use the notation $\sigma(i,i')$ to denote the successor of the pair $(i,i') \in \Delta$ in this ordering.

The Red Foundation

Our description of G' starts with the set of red vertices

$$R' = R_1 \cup \cdots \cup R_5$$

where these sets and their roles are described:

Gadgets to indicate k "chosen" vertices in G:

$$R_1 = \{a(i,u) : i \in [k],\ u \in V\}$$

Gadgets to indicate the intervals between chosen vertices:

$$R_2 = \{b(i,J) : i \in [k]^*,\ J \in \mathcal{J}'\}$$

Gadgets to indicate pairs of intervals:

$$R_3 = \{c(i,i',J,J') : i,i' \in [k]^*,\ J,J' \in \mathcal{J}'\}$$

Gadgets to indicate choice/interval pairs:

$$R_4 = \{c'(i,i',u,J) : i \in [k],\ i' \in [k]^*,\ u \in V,\ J \in \mathcal{J}'\}$$

Gadgets to indicate interval/choice pairs:

$$R_5 = \{c''(i,i',J,u) : i \in [k]^*,\ i' \in [k],\ J \in \mathcal{J}',\ u \in V\}$$

The red vertices of R' are organized into

$$k'' = 1 + 2k + 2k(k+1) + (k+1)^2$$

basic blocks defined as follows.

For $i = 1, ..., k$, let

$$\mathcal{A}(i) = \{a(i,u) : i \in [k],\ u \in V\}$$

For $i = 0, ..., k$, let

$$\mathcal{B}(i) = \{b(i,J) : i \in [k]^*,\ J \in \mathcal{J}'\}$$

For $(i,i') \in \Delta$, let

$$\mathcal{C}(i,i') = \{c(i,i',J,J') : J,J' \in \mathcal{J}'\}$$

For $i \in [k]$ and $i' \in [k]^*$ let

$$\mathcal{C}'(i,i') = \{c'(i,i',u,J) : u \in V,\ J \in \mathcal{J}'\}$$

For $i \in [k]^*$ and $i' \in [k]$ let

$$\mathcal{C}''(i, i') = \{c''(i, i', J, u) : J \in \mathcal{J}',\ u \in V\}$$

Note that these k'' blocks constitute a partition of the set R' of *basic* red vertices of G'. (Further blocks of red vertices will be added to G' as the construction progresses.)

Let $\mathcal{R}'$ denote the set of k'' blocks defined above.

The Semantics of the Reduction

Although we have only begun to describe G', enough is now visible to be able to partly describe the solution translations of the reduction. (This description may serve to motivate and clarify some of the further details of the proof.) The solution translations we refer to are:
(1) The manner in which a k-element perfect code in G translates to a k'-element perfect code in G' (the forward direction).
(2) The manner in which a k'-element perfect code (or odd dominating set) in G' translates to a k-element perfect code in G (the backward direction).

The Forward Solution Translation.

In the forward direction, suppose the k elements of a perfect code in G are

$$u_1 < u_2 < \cdots < u_k$$

Let $J_0 = J(0, u_1)$. In other words, J_0 is the interval of elements of V that precede u_1.

For $i = 1, ..., k-1$, let $J_i = J(u_i, u_{i+1})$. In other words, J_i is the interval of elements of V properly between u_i and u_{i+1}.

Let $J_k = J(u_k, \infty)$. Thus J_k is the interval of elements of V following u_k.

Our construction of G' will insure that the following set S of k'' red vertices extends to a perfect code in G'.

$$\begin{aligned} S &= \{a(i, u_i) : i \in [k]\} \\ &\cup \{b(i, J_i) : i \in [k]^*\} \\ &\cup \{c(i, i', J_i, J_{i'}) : i, i' \in [k]\} \\ &\cup \{c'(i, i', u_i, J_{i'}) : i \in [k],\ i' \in [k]^*\} \\ &\cup \{c''(i, i', J_i, u_{i'}) : i \in [k]^*,\ i' \in [k]\} \end{aligned}$$

The Backward Solution Translation.

Our construction of G' will also insure that any k'-element odd dominating set S in G' must be distributed so that there is exactly one element of S in each of the k'' red blocks of $\mathcal{R}'$. S will be forced to have the restricted form of a perfect code in G'. The construction will insure that for such an S the u-indices of the k elements of $S \cap R_1$ are distinct, and that these indicate a k-element perfect code in G.

Our semantic intentions in the design of G' can be summarized as follows:
(1) The k blocks of R_1 represent the k choices of a vertex of V that may form a k-element perfect code in G.
(2) The $k+1$ blocks of R_2 represent the intervals between the consecutive choices of vertices of V represented in R_1.
(3) The blocks of R_3, R_4 and R_5 do not really carry any new meaning with respect to G. They are principally concerned with the internal structure of G'.

The Blueprint for G'

We will describe G' by starting with the basic k'' red blocks of $\mathcal{R}'$ and applying various *operators* to these blocks. Each application of an operator results in further blocks of red and blue vertices being added to the construction, along with various edges.

In constructing G', operators are only applied to argument sets of red blocks. Thus each blue vertex β of G' is created by a single specific application of an operator, and the neighborhood of β is completely established by this application. This allows us to argue a series of claims concerning properties of G' as we continue to describe the steps of the construction.

The Block Guard Operator Γ_1. The operator Γ_1 takes as an argument a single red block $\mathcal{X}$ and adds one blue vertex connected to every red vertex in $\mathcal{X}$.

The last step of the construction of G' will be to apply this operator to each of the (by then) k' red blocks of the construction. We cannot make this the first step of the construction as some of the red blocks to which Γ_1 is to be applied have yet to be created by applications of operators in earlier steps of the construction. However, Γ_1 does provide a simple initial example of an operator and we can easily prove some important properties of G' having only this much information about the construction.

The Last Step: Apply the Γ_1 operator to every red block of the construction. (When we get to the last step, there will be k' red blocks, the collection of which will be denoted $\mathcal{R}$.)

Definition. Say that an application of an operator is *locally perfect* if the following condition is satisfied. Let A denote the union of the red blocks that either provide the arguments for the application of the operator, or that are created by the application of the operator. Let B denote the blue vertices created by the application of the operator. The condition is: if $S \subseteq A$ contains one vertex from each red block contained in A and if every vertex in B has at least one neighbor in S, then every vertex in B has a unique neighbor in S (in other words, S is "locally" a perfect code).

Claim 1. Any k'-element dominating set in G' must contain exactly one vertex in each of the k' blocks of $\mathcal{R}$, and the last step of the construction of G' is locally perfect.

Proof. Let S be a k'-element dominating set in G'. If any block does not contain an element of S, then the blue vertex created for this block by Γ_1 is not dominated by S. Since each of the k' blocks of G' must contain at least one element of S, each block must contain exactly one element of S. Local perfection is obvious. □

Claim 2. Every dominating set of G' has size at least k'.

Proof. This also follows from the proof of Claim 1. □

The Branch Operator Γ_2. The arguments to the Γ_2 operator are two red blocks $\mathcal{X}$ and $\mathcal{X}'$ given with an assignment to each element of $\mathcal{X}$ of a nonempty proper subset of $\mathcal{X}'$:

$$x \in \mathcal{X} \;\mapsto\; S_x \subseteq \mathcal{X}'$$

The result of applying Γ_2 to $\mathcal{X}$ and $\mathcal{X}'$ with this assignment is:
(1) The creation of the set of $|\mathcal{X}|$ blue vertices

$$\{\beta(x) : x \in \mathcal{X}\}$$

(2) For each blue vertex $\beta(x)$ in this set, the creation of edges connecting $\beta(x)$ to $x \in \mathcal{X}$ and to each red vertex in $\mathcal{X}' \setminus S_x$.

Claim 3. Suppose the operator Γ_2 is applied to the red blocks $\mathcal{X}$ and $\mathcal{X}'$, and suppose S is a k'-element dominating set in G', with $S \cap \mathcal{X} = \{x\}$. Then $S \cap \mathcal{X}' = \{x'\}$ with $x' \in S_x$, and the application of Γ_2 is locally perfect.

Proof. This follows almost immediately from the definition of Γ_2. □

A Special Case: Γ_2 as Equality Operator A useful special case application of Γ_2 is as an "equality" enforcer. For this, assume that the arguments to Γ_2 are isomorphic sets, with the isomorphism given by the correspondence:

$$x \in \mathcal{X} \;\longleftrightarrow\; e(x) \in \mathcal{X}'$$

and in applying the operator, we use the assignment given by:

$$x \;\mapsto S_x = \{e(x)\}$$

In this situation, by Lemma 4, and assuming that k' is an odd dominating set in G', if $S \cap \mathcal{X} = \{x\}$ and $S \cap \mathcal{X}' = \{y\}$, then $y = e(x)$.

Step 1(a): For $i = 1, ..., k-1$, apply the operator Γ_2 to the pair of arguments:

$$\mathcal{X} = \mathcal{A}(i) \quad \text{and} \quad \mathcal{X}' = \mathcal{B}(i)$$

with the assignment:

$$a(i,u) \;\mapsto\; \{b(i,J) : J \in \mathcal{J}',\ \partial(J) = u\}$$

Step 1(b): Apply the operator Γ_2 to the arguments:

$$\mathcal{X} = \mathcal{A}(k) \quad \text{and} \quad \mathcal{X}' = \mathcal{B}(k)$$

with the assignment:

$$a(k,u) \;\mapsto\; \{b(k,J) : J \in \mathcal{J}',\ \partial(J) = u \text{ and } \partial'(J) = \infty\}$$

Step 2(a): For $i = 1, ..., k-1$, apply the operator Γ_2 to the pair of arguments:

$$\mathcal{X} = \mathcal{A}(i+1) \quad \text{and} \quad \mathcal{X}' = \mathcal{B}(i)$$

with the assignment:

$$a(i+1,u) \;\mapsto\; \{b(i,J) : J \in \mathcal{J}',\ \partial'(J) = u\}$$

Step 2(b): Apply the operator Γ_2 to the arguments:

$$\mathcal{X} = \mathcal{A}(1) \quad \text{and} \quad \mathcal{X}' = \mathcal{B}(0)$$

with the assignment:

$$a(1,u) \;\mapsto\; \{b(0,J) : J \in \mathcal{J}',\ \partial'(J) = u \text{ and } \partial(J) = 0\}$$

Claim 4. Suppose that S is a k'-element dominating set in G'. Then the $2k+1$ vertices of $S \cap (R_1 \cup R_2)$ consistently represent k vertices of G and the intervals between them in the following way:
(1) For $1 \le i < i' \le k$, if $S \cap \mathcal{A}(i) = \{a(i,u)\}$ and $S \cap \mathcal{A}(i') = \{a(i',u')\}$, then $u < u'$.
(2) For $i = 1, ...k-1$, if $S \cap \mathcal{A}(i) = \{a(i,u)\}$ and $S \cap \mathcal{A}(i+1) = \{a(i+1,v)\}$, then $S \cap \mathcal{B}(i) = \{b(i, J(u,v))\}$.
(3) If $S \cap \mathcal{A}(1) = \{a(1,u)\}$, then $S \cap \mathcal{B}(0) = \{b(0, J(0,u))\}$.
(4) If $S \cap \mathcal{A}(k) = \{a(k,u)\}$, then $S \cap \mathcal{B}(k) = \{b(k, J(u,\infty))\}$.

Proof. This follows from Claim 1 and Steps 1 and 2, noting that the blue vertices created in these steps prevent any inconsistencies with the specifications of the claim. □

The Product Operator Γ_3. The arguments for the operator are three red blocks: $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{Z}$, together with a 1:1 product correspondence:

$$z \in \mathcal{Z} \;\longleftrightarrow\; (z_1, z_2) \in \mathcal{X}_1 \times \mathcal{X}_2$$

An application of Γ_3 augments G' in the following ways:
(1) Two blocks of red vertices are created:

$$\begin{aligned} \mathcal{P}_1 &= \{\rho_1(x_1,x_2) : x_1 \in \mathcal{X}_1,\ x_2 \in \mathcal{X}_2\} \\ \mathcal{P}_2 &= \{\rho_2(x_1,x_2) : x_1 \in \mathcal{X}_1,\ x_2 \in \mathcal{X}_2\} \end{aligned}$$

(2) The operator Γ_2 is applied to the arguments

$$\mathcal{X} = \mathcal{X}_1 \text{ and } \mathcal{X}' = \mathcal{P}_1$$

with the assignment

$$x \in \mathcal{X}_1 \;\mapsto\; \{\rho_1(x,y) : y \in \mathcal{X}_2\}$$

(3) The operator Γ_2 is applied to the arguments

$$\mathcal{X} = \mathcal{X}_2 \text{ and } \mathcal{X}' = \mathcal{P}_2$$

with the assignment

$$y \in \mathcal{X}_2 \;\mapsto\; \{\rho_2(x, y) : x \in \mathcal{X}_1\}$$

(4) The operator Γ_2 is applied as an equality operator to the arguments

$$\mathcal{X} = \mathcal{P}_1 \text{ and } \mathcal{X}' = \mathcal{P}_2$$

with the assignment

$$\rho_1(x_1, x_2) \;\mapsto\; \{\rho_2(x_1, x_2)\}$$

(5) The operator Γ_2 is applied as an equality operator to the arguments

$$\mathcal{X} = \mathcal{P}_2 \text{ and } \mathcal{X}' = \mathcal{Z}$$

with the assignment

$$\rho_2(z_1, z_2) \;\mapsto\; \{z\}$$

(6) The operator Γ_2 is applied as an equality operator to the arguments

$$\mathcal{X} = \mathcal{Z} \text{ and } \mathcal{X}' = \mathcal{P}_1$$

with the assignment

$$z \;\mapsto\; \{\rho_1(z_1, z_2)\}$$

Claim 5. Suppose the operator Γ_3 has been applied to the red blocks $\mathcal{X}_1, \mathcal{X}_2$ and $\mathcal{Z}$, and suppose S is a k'-element dominating set in G', with $S \cap \mathcal{X}_1 = \{x\}$ and $S \cap \mathcal{X}_2 = \{y\}$. Then $S \cap \mathcal{Z}$ contains the single element z for which $z_1 = x$ and $z_2 = y$, and the application of Γ_3 is locally perfect.

Proof. By Claim 1, $S \cap \mathcal{Z}$ contains just one element z. Suppose $z_1 = p$ and $z_2 = q$. Suppose

$$S \cap (\mathcal{P}_1 \cup \mathcal{P}_2) = \{\rho_1(x', y'), \rho_2(x'', y'')\}$$

By Claim 3 and the application of Γ_2 in (2) and (3) of the definition of Γ_3, $x' = x$ and $y'' = y$. By Claim 3 and the application of Γ_2 in (4), $x' = x''$ and $y' = y''$ and therefore $x'' = x$. By Claim 3 and the application of Γ_2 in (5), $x'' = p$ and $y'' = q$ and therefore $p = x$ and $q = y$. Since an application of Γ_3 is made up of applications of Γ_2, the local perfection of Γ_3 follows from Claim 3. $\square$

Step 3: For each pair (i, i') with $i, i' \in [k]^*$, apply the operator Γ_3 to the arguments: $\mathcal{X}_1 = \mathcal{B}(i)$, $\mathcal{X}_2 = \mathcal{B}(i')$ and $\mathcal{Z} = \mathcal{C}(i, i')$ with the product correspondence:

$$c(i, i', J, J') \in \mathcal{C}(i, i')\mathcal{B}(i) \times \mathcal{B}(i')$$

Step 4: For each pair (i, i') with $i \in [k]$ and $i' \in [k]^*$, apply the operator Γ_3 to the arguments: $\mathcal{X}_1 = \mathcal{A}(i)$, $\mathcal{X}_2 = \mathcal{B}(i')$ and $\mathcal{Z} = \mathcal{C}'(i, i')$ with the product correspondence:

$$c'(i, i', u, J') \in \mathcal{C}'(\delta) \;\longleftrightarrow\; (a(i, u), b(i', J)) \in \mathcal{A}(i) \times \mathcal{B}(i')$$

Step 5: For each pair (i, i') with $i \in [k]$ and $i' \in [k]^*$, apply the operator Γ_3 to the arguments: $\mathcal{X}_1 = \mathcal{B}(i)$, $\mathcal{X}_2 = \mathcal{A}(i')$ and $\mathcal{Z} = \mathcal{C}''(i, i')$ with the product correspondence:

$$c''(i, i', J, u') \in \mathcal{C}''(i, i')(b(i, J), b(i', u')) \in \mathcal{B}(i) \times \mathcal{A}(i')$$

The G-Adjacency Operator Γ_4. This operator takes as arguments all of the $\mathcal{A}$ blocks. A set of n blue vertices $\{\beta(u)\}$ is created. Each vertex $\beta(u)$ is connected to each red vertex in the set $\{a(i, v) : v \in N[u]\}$. (Thus if $|N[u]| = m$ in G, then $\beta(u)$ has km red neighbors in G'.)

Step 6: Apply the Γ_4 operator once (only) to all of the $\mathcal{A}$ blocks (i.e., to the vertices of R_1).

Claim 6. If S is a k'-element dominating set in G', then the V-coordinates of the vertices of $S \cap R_1$ indicate a k-element dominating set in G. Furthermore, S is a perfect code in G', then the V-coordinates of $S \cap R_1$ indicate a k-element perfect code in G.

Proof. It follows from Claim 4 that the V-coordinates of $S \cap R_1$ indicate k distinct vertices of G. The rest follows from the definition of Γ_4. $\square$

The Perfection Operator Γ_5. This operator takes as arguments all of the $\mathcal{C}, \mathcal{C}'$ and $\mathcal{C}''$ blocks. An application of the operator results in:
(1) The creation of a set of blue vertices:

$$\{\beta(u, v) : \ u \text{ is close to } v \text{ in } G\ \}$$

(2) The creation of edges connecting each blue vertex $\beta(u, v)$ in this set to each red vertex in the set

$$\begin{aligned} &\{c(i, i', J, J') : i, i' \in [k]^*,\ u \in J,\ v \in J'\} \\ &\cup \{c'(i, i', u, J) : i \in [k],\ i' \in [k]^*,\ v \in J\} \\ &\cup \{c''(i, i', J, v) : i \in [k]^*,\ i' \in [k],\ u \in J\} \end{aligned}$$

Step 7: Apply the operator Γ_5 once (only) to all of the $\mathcal{C}, \mathcal{C}'$ and $\mathcal{C}''$ blocks.

Claim 7. If S is a k'-element dominating set in G', then:
(1) Each blue vertex created in Step 7 is adjacent to at most one vertex of S and Step 7 is locally perfect.
(2) The V-coordinates of $S \cap (R_1 \cup R_2)$ represent a set of k vertices in G such that the distance in G between any two of these k is at least 3.

Proof. By Claim 4, $S \cap (R_1 \cup R_2)$ consistently represents k vertices of G and the k intervals between them. Each vertex of V is in exactly one of these $2k$ sets. Thus for each ordered pair of vertices (u, v) of G, exactly one statement is true on the following list, describing where u is to be found and where v is to be found:

- u is in the ith interval and v is in the jth interval.

- u is the ith chosen vertex and v is in the jth interval.
- u is in the ith interval and v is the jth chosen vertex.
- u is the ith chosen vertex and v is the jth chosen vertex.

All but the last of these statements (which is unacceptable for a close pair in a perfect code) is reflected in the edges added to G' in Step 7. By Claim 5, the information represented by $S \cap (R_3 \cup R_4 \cup R_5)$ is consistent with the information represented by $S \cap (R_1 \cup R_2)$ according to the semantics of the reduction. This yields (1); (2) follows from this discussion and the definition of Γ_5, since if there were a close pair of chosen vertices, there would be a blue vertex created in Step 7 with no neighbor in S. □

Claim 8. If S is a k'-element dominating set in G', then then V-coordinates of $S \cap R_1$ represent a k-element perfect code in G.

Proof. This follows from Claims 4, 6 and 7. □

Step 8: The Last Step. Apply the Γ_1 operator to every red block of the construction thus far.

Claim 9. If S is a k'-element dominating set in G', then S is actually a perfect code in G'.

Proof. This follows from Claims 1, 3, 5, 6 and 7. □

The proof of the theorem can now be completed. The difficult half of (1) is established by Claim 8, (2) is proved by Claim 2 and (3) is the same as Claim 9. The reader can now check the few remaining details to see that the forward translation of a k-element perfect code in G to a k'-element perfect code in G' works correctly. □

4 Membership in $W[2]$

In this section we sketch how the problems Maximum Likelihood Decoding, Weight Distribution and Minimum Distance can be shown to belong to the parameterized complexity class $W[2]$. For this argument, we depend upon the improved characterization of $W[2]$ established in [DF96], where it is shown that the following variant of Satisfiability is complete for $W[2]$.

(n, k, n, c)-Weighted Satisfiability
Instance: A boolean expression E that is an n-product of $f(k)$-sums of n-products of sums of size at most c, where c is a fixed constant and f is an arbitrary but fixed function.
Parameter: k.
Question: Is there a truth assignment of weight k that satisfies E?

Note that the parameter plays two different roles in this problem: in the structure of the expression E and in the weight of the truth assignment. (A truth assignment has *weight* k if it assigns exactly k variables the value *true*, and all others the value *false*.)

We sketch the argument for Maximum Likelihood Decoding. This immediately implies

$W[2]$ membership for the special case of WEIGHT DISTRIBUTION and the argument can be easily adapted to prove that MINIMUM DISTANCE is in $W[2]$.

Given an instance (H, s, k) of MAXIMUM LIKELIHOOD DECODING we create k blocks of variables, one for each "choice" of a binary vector $x_i \in H$ for which possibly:

$$\sum_{i=1}^{k} x_i = s$$

It is easy (and a standard technique [DF95a, DF95b]) to produce a weft 1 expression E_1 that is satisfiable only if exactly one variable in each of the choice blocks is set to true.

To express that the sum of the x_i is s, we require that for each index j from 1 to the length n of the vectors in H, there is some specification σ of precisely which of the k chosen vectors is 1 in this component (the number of such being even or odd, depending on $s[j]$ is 0 or 1). Thus there are at most $2^k/2$ possible (even or odd) specifications σ for a given $s[j]$. To each such σ (for $s[j]$) we can associate a (large) product of size 2 sums to enforce that the k choices of the x_i are compatible with the specification σ for $s[j]$. This discussion translates directly into an expression E_2 of the form admitted by (n, k, n, c)-WEIGHTED SATISFIABILITY, and we take $E = E_1 \wedge E_2$ which also has this form.

5 Concluding Discussion and Open Problems

The obvious outstanding open problems are:
(1) Is the MINIMUM DISTANCE problem, recently proved to be *NP*-complete by Vardy [Var97], also hard for $W[1]$?
(2) Is the SHORTEST VECTOR problem hard for *NP* and $W[1]$?

If the following graph-theoretic transformation analogous to Theorem 1 can be found, then both of these questions can be answered in the affirmative.

Transformation $1'$**.** Let $G = (R, B, E)$ be a red/blue graph on n vertices, and k a positive integer. In time polynomial in n and k we can produce a red/blue graph $G' = (R', B', E')$ and a positive integer k' that is purely a function of k such that:
(1) G has an even set of size k if and only if G' has a an even set of size k'.
(2) G has a **Z**-even set of size k if and only if G' has a **Z**-even set of size k'.
(3) Every even set in G' has size at least k'.
(4) Every **Z**-even set in G' has size at least k'.

Proposition 8. The existence of Transformation $1'$ implies that MINIMUM DISTANCE is hard for $W[1]$ and *NP*-complete.

Proof. We reduce from EXACT EVEN SET, shown to be hard for $W[1]$ in Theorem 5. Suppose the instance (G, k) is transformed by the means of Transformation $1'$ to (G', k'). Let $\mathcal{V}$ be the set of vectors corresponding to the red/blue adjacency matrix of G'.

If G has a k-element even set, then by (1) of Transformation $1'$, G' has an even set R' of size k'. The vectors in $\mathcal{V}$ corresponding to the vertices of R' sum to the all-zero vector over $GF[2]$.

Conversely, suppose $\mathcal{V}$ has a subset S of size at most k' such that the vectors of S sum to the all-zero vector over $GF[2]$. Let R_S be the corresponding set of red vertices in G'. It is immediate that R_S is an even set in G'. By (3) of Transforamtion $1'$, $|R_S| = k'$. By (1) of Transformation $1'$, G has a k-element even set. $\square$

Proposition 9. The existence of Transformation $1'$ implies that Shortest Vector is hard for NP and $W[1]$.

Proof. We reduce from the problem Exact **Z**-Even Set, shown to be NP and $W[1]$ hard in Theorem 6. Let (G, k) be an instance of that problem and let (G', k') be given by Transformation $1'$, $G' = (R', B', E')$.

Identify R' with $\{1, ..., |R'|\}$. Let $\delta^1(u)$ denote the vector of length $|R'|$ that has the value 2 in the uth component, and is otherwise 0. Choose $\ell > k'$. Let $\delta^{\ell^2}(j)$ denote the vector of length $|B'|$ that is ℓ^2 in the jth component and otherwise 0.

The instance (L, w) of Shortest Vector to which we transform (G', k') is described:
(1) The basis of the lattice L is given by the set of vectors $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ where

$$\begin{aligned} \mathcal{V}_1 &= \{x_u^\ell \cdot \delta^1(u) : u \in R'\}\} \\ \mathcal{V}_2 &= \{\delta^{\ell^2}(j) \cdot \mathbf{0}^{|R'|} : 1 \leq j \leq |B'|\} \end{aligned}$$

(2) The parameter is $w = k'$.

The basis for L described above consists of $|R'| + |B'|$ vectors of length $|R'| + |B'|$ that are clearly linearly independent set over the reals.

If G has a k-element **Z**-even set, then by (2) of Transformation $1'$, G' has a k'-element **Z**-even set. The corresponding vectors of $\mathcal{V}$ therefore have a **Z**-linear combination x with $||x||^2 = k' = w$.

Conversely, suppose S is a set of vectors of $\mathcal{V}$ such that there is a **Z**-linear combination x of the elements of S (with non-zero coefficients) such that $||x||^2 = w$.

Claim. None of the vectors of $\mathcal{V}_2$ belongs to S.

Proof. The first $|B'|$ components of x are necessarily multiples of ℓ. Since $\ell^2 > w$, it must be the case that x is 0 in these components. If S were to contain a vector in $\mathcal{V}_2$, then the absolute value of the sum of the coefficients multiplying the elements of S that belong to $\mathcal{V}_1$ must be at least ℓ. This implies that $||x||^2 \geq \ell^2 > w$.

It follows from the claim that each vector in S corresponds to a vertex in G'. Let R_S denote the corresponding set of vertices in G'. R_S is a **Z**-even set in G', and $|R_S| \leq k'$ because of the bound on $||x||^2$. By (4) of Transformation $1'$, $|R_S| = k'$. By (2) of Transformation $1'$, G has a

$\mathbf{Z}$-even set of size k. $\square$

References

[ABSS93] S. Arora, L. Babai, J. Stern and Z. Sweedyk. The hardness of approximate optima in lattices, codes and systems of linear equations. *Proc. 34th Annual Symp. Found. Computer Science* (1993), 724–733.

[Aj96] M. Ajtai. Generating hard instances of lattice problems. *Proc. 28th Annual ACM Symp. Theory of Computing* (1996), 99–108.

[BFR96] R. Balasubramanian, M. Fellows and V. Raman. An improved fixed-parameter algorithm for vertex cover. *To appear.*

[BFH94] H. Bodlaender, M.R. Fellows and M.T. Hallett. Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy. *Proc. ACM Symp. on Theory of Computing (STOC)* (1994), 449–458.

[BMcEvT78] E.R. Berlekamp, R.J. McEliece and H.C.A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory* 24 (1978), 384–386.

[LeC97] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. Tecnical Report, Department of Computer Science, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. To appear in *Information Processing Letters.*

[CC93] L. Cai and J. Chen. Fixed parameter tractability and approximability of NP-hard optimization problems. *Proc. 2nd Israel Symp. on Theory and Computing Systems* (1993), 118–126.

[CCDF96] L. Cai, J. Chen, R.G. Downey and M.R. Fellows. On the parameterized complexity of short computation and factorization. *Arch. for Math. Logic*, to appear.

[CCDF97] L. Cai, J. Chen, R.G. Downey and M.R. Fellows. Advice classes of parameterized tractability. *Annals of Pure and Applied Logic* (1997), to appear.

[DF95a] R.G. Downey and M.R. Fellows. Fixed parameter tractability and completeness I: basic theory. *SIAM Journal of Computing* 24 (1995), 873–921.

[DF95b] R.G. Downey and M.R. Fellows. Fixed parameter tractability and completeness II: completeness for $W[1]$. *Theoretical Computer Science A* 141 (1995), 109–131.

[DF95c] R.G. Downey and M.R. Fellows. Parameterized computational feasibility. In: *Feasible Mathematics II, P. Clote and J. Remmel (eds.)* Birkhauser, Boston (1995) 219–244.

[DF96] R.G. Downey and M.R. Fellows. Threshold dominating sets and an improved characterization of $W[2]$. To appear: *Theoretical Computer Science A.*

[DF97] R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1997, to appear.

[DFKHW94] R.G. Downey, M.R. Fellows, B.M. Kapron, M.T. Hallett and H.T. Wareham. The parameterized complexity of some problems in logic and linguistics. *Proceedings Symposium on Logical Foundations of Computer Science (LFCS)*, Springer-Verlag, Lecture Notes in Computer Science vol. 813 (1994), 89–100.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.

[Kan87] R. Kannan. Minkowski's convex body problem and integer programming. *Mathematics of Operations Research* 12 (1987).

[KK88] J. Kratochvíl and M. Křivánek. On the computational complexity of codes in graphs. In: *Proceedings MFCS Karlovy Vary 1988*, Lecture Notes in Comp. Sci. 324, Springer-Verlag, Berlin 1988, pp 396–404.

[Kra94] J. Kratochvil Regular codes in regular graphs are difficult. *Discrete Mathematics* 133 (1994), 191–205.

[KST94] H. Kaplan, R. Shamir and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: minimum fill-in and DNA physical mapping (extended abstract). In *35th Ann. Proc. of the Foundations of Computer Science (FOCS '94)*, (1994), 780–891.

[Lov86] L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity.* NSF-CBMS Conference Series, SIAM, 1986.

[St92] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classification,* Vol. 9 (1992), 91-116.

[Var97] A. Vardy. Algorithmic complexity in coding theory and the minimum distance problem. To appear in: *Proc. ACM Symposium on the Theory of Computing*, 1997.

The intractability of computing the minimum distance of a code. *Manuscript, Nov., 1996.*

[vEB80] P. van Emde Boas. Another *NP*-complete partition problem and the complexity of computing short vectors in a lattice. *Technical Report* 81-04, Dept. of Mathematics, Univ. Amsterdam, 1980.

Appendix 1: Background on Parameterized Complexity

It has become increasingly clear that classical complexity frameworks such as *NP*-completeness and *PSPACE*-completeness are not adequate to address intractability questions for problems that are naturally parameterized, and for which important applications are covered by $k \leq 50$.

For example, the Vertex Cover problem is one of the six basic *NP*-complete problems of Garey and Johnson [GJ79]. The best known result presently is that this can be solved in time $O(kn + (4/3)^k k^2)$ with a very small hidden constant [BFR96]. This means that although the problem is *NP*-complete, it is well-solved for input graphs of *unlimited* size, so long as k is at most 70. Strong fixed-parameter tractability results such as this seem to be not uncommon or beyond reasonable hope when problems are qualitatively classifiable as belonging to *FPT*. Three of the six basic *NP*-complete problems of Garey and Johnson [GJ79] are *fixed-parameter tractable* (formally defined below).

In general, a variety of metrics can be applied to the input of a computational problem — the total length of the input is one basic measurement, but it is by no means the only important one. It is natural to try to understand how different input measurements interact in determining problem complexity, and it is essential to understand these kinds of interactions in order to exploit the opportunities for designing algorithms that are sensitive to natural input distributions.

A generic example of a parameterization is provided by the many well-known decision problems concerning graphs that take as input a graph G and a positive integer k.

The parameter k appears to contribute to the complexity of such graph problems in two qualitatively distinct ways. Graph Genus, Min Cut Linear Arrangement, Vertex Cover and Feedback Vertex Set for Undirected Graphs (see [GJ79] for definitions) can all be solved in time $O(f(k)n^c)$ where c is a constant independent of k and f is some (arbitrary) function. This "good behavior" is termed *fixed-parameter tractability* in the theory introduced in [DF95a]. As is the case with polynomial-time complexity, the exponent c is typically small. One can equivalently define fixed-parameter tractability to be solvability in time $O(f(k) + n^c)$, i.e., with only an *additive* exponential contribution from the parameter [CCDF97]. There is a rich collection of distinctive techniques for devising *FPT* algorithms (see [DF95c, DF97, KST94, LeC97, St92]).

Contrasting complexity behavior is exhibited by the naturally parameterized problems Clique, Dominating Set and Bandwidth, for which the best known algorithms have running times $O(n^{ck})$. These problems have been shown to be complete or hard for various levels of the W hierarchy of parameterized complexity

$$W[1] \subseteq W[2] \subseteq \cdots \subseteq W[P] \subseteq \cdots XP$$

and this can be taken as evidence that they are unlikely to be fixed-parameter tractable [BFH94]. With these problems, we seem to hit a natural "wall" requiring brute force effort, much as is typically the case with *NP*-complete problems. For example, essentially no better algorithm is known for the k-Dominating Set problem than checking all k-subsets.

As in the theory of *NP*-completeness, there are roughly two kinds of arguments that can be offered for believing that parameterized problems that are complete or hard for $W[1]$ are not likely to be fixed-parameter tractable. The first kind of argument is, roughly speaking, sociological: so many different kinds of problems stand or fall together that the combination of efforts expended unsuccessfully from the various vantages compels a belief in inherent intractability. The second kind of argument is some form of direct intuition concerning the nature of the computations that define the issue (e.g., nondeterministic as opposed to deterministic P-time).

For parameterized complexity, both kinds of arguments can be made. Although the amount of unsuccessful effort that has been expended in attempts to show fixed-parameter tractability for $W[1]$-hard problems is much less than the total effort expended to date in attempting to develop polynomial-time algorithms for NP-complete problems, it is still considerable and accumulating. The parameterized complexity class $W[1]$ is particularly interesting and important, for several reasons. Indeed, there is a substantial list of very natural and important problems that are precisely $W[1]$-complete, including: Clique, Independent Set, Vapnik-Chervonenkis Dimension, Monotone Data Complexity for Relational Databases, Square Tiling, k-Step Derivation for Context Sensitive Grammars, m-Length Common Subsequence for k Sequences, and k-Length Post Correspondence.

Direct intuition about $W[1]$ is also available. It is shown in [DFKHW94, CCDF96] that the k-step Halting Problem for nondeterministic Turing machines (Short Turing Machine Acceptance), is $W[1]$-complete. This problem is formally defined as follows.

Short Turing Machine Acceptance
Instance: A nondeterministic Turing machine M and a positive integer k.
Parameter: k
Question: Does M have a computation path accepting the empty string in at most k steps?

This is a problem so generic and opaque that it is hard to imagine that there is any algorithm for it that radically improves on simply exploring the n-branching depth k tree of allowed transitions exhaustively. (This is essentially the same intuition as the belief that Cook's Theorem provides a basis for the intractability of NP complete problems.)

For the definitions of the $W[t]$ classes and the fundamentals of parameterized complexity, we refer the reader to [DF95a, DF95b, DF95c, DF97]. In order to work with the theory we need only the following basic definitions.

Definition. A *parameterized problem* is a set $L \subseteq \Sigma^* \times \Sigma^*$ where Σ is a fixed alphabet. For convenience, we consider that a parameterized problem is a subset $L \subseteq \Sigma^* \times N$.

Definition. We say that a parameterized problem L is (uniformly) *fixed-parameter tractable* if there is a constant α and an algorithm Φ such that Φ decides if $(x,k) \in L$ in time $f(k)|x|^\alpha$ where $f: N \to N$ is an arbitrary function.

Definition. Let A, B be parameterized problems. We say that A is (uniformly many:1) *reducible* to B if there is an algorithm Φ which transforms (x,k) into $(x', g(k))$ in time $f(k)|x|^\alpha$, where $f, g : N \to N$ are arbitrary functions and α is a constant independent of k, so that $(x,k) \in A$ if and only if $(x', g(k)) \in B$.

It is easy to see that if A reduces to B and B is fixed parameter tractable then so too is A.