Information Processing Letters, vol. 42, no. 3, May 1992, pp. 121-126

Complexity of the Closest Vector Problem in a Lattice Generated by (0,1)-Matrix*

Boleslaw K. Szymanski and Balaram Sinharoy

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, N.Y. 12180-3590, USA

Keywords: NP-completeness, closest vector, lattice, satisfiability.

1 Introduction

The *closest vector problem*, often referred to as CVP, is to find a vector in a lattice that is closest to a given (input) vector. The problem, well-known in mathematical programming, was proven NP-hard for an arbitrary lattice and norm [1, 6]. This general result does not apply, however, to problems in which a lattice is generated by a matrix whose elements belong to some proper subset of integers. For example, minimization problems in graphs and networks can be treated as CVP in a lattice generated by the incidence matrix of a graph or a network. Incidence matrices have their elements in $\{0, 1\}$ or $\{-1, 0, 1\}$ only, therefore the complexity of such problems cannot be established on the basis of the general result mentioned above.

In this paper we investigate complexity of the CVP in a lattice generated by the matrix whose elements are in $\{0, 1\}$. We refer to this problem as (0,1)-CVP. Our interest in (0,1)-CVP was motivated by the data alignment problem for SIMD architectures [5]. In a Single Instruction stream – Multiple Data stream (SIMD) computer, a significant speedup can be achieved by distributing (or mapping) data structures in a program to individual processors. One processor is allocated (at least conceptually) per data item of an array or other composite data structures. Operations involving arrays can be performed entirely locally if the operand array elements are allocated to the same processor. Otherwise, computation has to be interwoven

*This work was partially supported by National Science Foundation under grants CCR-8920694 and CDA-8805910. Some results of this paper were presented at XIV International Symposium on Mathematical Programming, Amsterdam, Netherlands, August 5–9, 1991. The paper was published in *Information Processing Letters*, vol. 42, May 1992, pp. 121-6,

with a time consuming interprocessor communication needed to fetch non-local arguments and to store non-local results. The cost of communication depends on network locations of the communicating processors and the network interconnection pattern. One of the major challenges in programming SIMD computers is to distribute data structures among the processors in such a way that the interprocessor communication is minimized. The *data alignment* problem [5] is to determine the relative shifts of the distributed data structures' dimensions so that the required interprocessor communication is at minimum.

The alignment problem is equivalent to finding the closest vector in a lattice generated by a matrix with elements in {-1,0,1} [5]. For hypercubes and the four nearest neighbor mesh, the two networks often used in practice, the distance between processors is measured in the first norm. In the eight nearest neighbor interconnection the corresponding norm is infinity. As shown later, the (0,1)-CVP (equivalent to the data alignment problem) is NP-hard in these norms, except in the (trivial) case of infinity norm and the input vector with elements in $\{0, 1\}$. The NP-hardness of (0,1)-CVP may be useful in establishing complexity of other minimization problems defined over (0,1)-matrices, in particular, incidence matrices of graphs and networks which are frequently encountered in applications.

The paper is organized as follows. The next section provides the basic definitions. The proof that (0,1)-CVP is NP-hard in an arbitrary finite norm is given in Section 3. Complexity of (0,1)-CVP in infinity norm is analyzed in Section 4. If the input vector elements are in $\{-1, 0, 1\}$ then (0,1)-CVP has a (trivial) polynomial-time solution. In all other cases it is NP-hard. Unlike the previously published proofs for the general CVP (Kannan [1] used three dimensional matching whereas van Emde Boas [6] used partition), reductions in both sections use 3SAT problem or its variant NOT-ALL-EQUAL [3]. Both sections use the same method of reducing a propositional logic problem to a satisfiability of inequality over integers, which in turn is reduced to the relevant minimization problem. This method might be useful in analyzing complexity of other minimization problems. The last section provides final remarks about the techniques used in the presented proofs and about possible applications of the results.

2 Basic Definitions

The set $L(A) = \{Y \in \Re^p : Y = AX, X \in Z^q\}$, where A is a $p \times q$ matrix with elements in $\Re$ and linearly independent columns, is called a *lattice* generated by the columns of A or, in short, a lattice based on matrix A [4]. In other words, a *lattice* L in $\Re^p$ is the set of all *integer linear combinations* of a set of linearly independent vectors (i.e. columns of A) in $\Re^p$.

The independent vectors are called a *basis* of the lattice. The *dimension* of the lattice is the number of basis vectors (the number of columns in A) that generate it. A *distance* between two vectors $C = [c_1, \ldots, c_m]$ and $Y = [y_1, \ldots, y_m]$ in the l-th norm is defined as:

$$||C - Y||_l = \sqrt[l]{\sum_{i=1}^{m} |c_i - y_i|^l}$$

In infinity norm the distance definition simplifies to:

$$||C - Y||_\infty = \max_{i=1,\ldots,m} |c_i - y_i|$$

The *closest vector problem*, abbreviated here as CVP, is to find the vector in a lattice that is closest to the given input vector C, i.e. to find a vector $Y_{min} \in L(A)$ such that $||C - Y_{min}||_l \leq ||C - Y||_l$ for all vectors $Y \in L(A)$. The variant of CVP in which a lattice is based on a matrix with elements in $\{0, 1\}$ is referred to here as (0,1)-CVP.

Two NP-hard problems in propositional logic used in reductions below are 3SAT and its variant NOT-ALL-EQUAL [3]. Let U be a set of boolean variables, and CL be a collection of clauses over U such that each clause is a disjunction of exactly three literals. Each literal is either a variable $x \in U$ or the boolean negation $\bar{x}$ of such a variable. The 3SAT problem is to determine if there is a truth assignment for the variables in U such that each clause in CL has at least one true literal. The NOT-ALL-EQUAL problem is to establish whether there is a truth assignment for the variables in U which yields at least one pair of unequal literals in each clause in CL.

3 (0,1)-CVP Complexity in Finite Norms

For a finite norm l and a given integer constant $r \neq 0$, we reduce the NOT-ALL-EQUAL problem to the (0,1)-CVP. First, we replace each boolean variable in the original problem by a quadruplet of arithmetic variables to represent: the boolean variable itself, its boolean negation and the arithmetic negations of the first two variables. Then, we define inequalities over these arithmetic variables such that a solution to the inequalities exists if and only if there is the truth assignment that satisfies the original problem. Introduced inequalities must have a form suitable for a distance definition that is expressed as a sum of l-th powers of absolute values of some linear combinations of the point coordinates. Moreover, the elements of sought matrix A are in $\{0, 1\}$ and input vector elements c_i are in $\{0, r\}$. Hence, coefficients of variables that appear inside each argument of absolute value have to be "+1" and a free term, if any, must be "$-r$". Even such limited form is sufficient to duplicate a given value, say v, by using inequality $|v+\bar{x}|^l+|\bar{x}+x|^l \leq 0$ (its only solution is $x = v = -\bar{x}$). Likewise, inequality $|v + x_0|^l + |v + x_1|^l + |x_2 + x_0 + x_1|^l \leq 0$ creates a double of v in x_2, while inequality $|v+y_0+y_1|^l+|y_2+y_1|^l+|y_2+y_0|^l \leq 0$ has as solution $y_2 = v/2$ (but only if v is even). We want to represent boolean values *True, False* by arithmetic values $h, h + 1$. The nice property of such representation is that sum of representation of any boolean value and its negation is a constant $2h+1$ while difference of them has constant absolute value 1. It is also important that for any pair of integers, say x, y, if $x + y - (2h + 1) = 0$ then $\min |x - y| = 1$. If constant r in (0,1)-CVP is odd, it can be readily used as $2h + 1$, otherwise we need to "divide out" all factors of two from r. Finally, it is easy to observe that three boolean values compared pairwise yield three equalities, if all three are equal, and one equality, otherwise. This observation leads to a short inequality (cf. inequality (6)) whose satisfiability is equivalent to the existence of the required truth assignment for NOT-ALL-EQUAL problem.

In summary, the NOT-ALL-EQUAL problem is reduced first to a problem F, defined below, expressed in terms of arithmetic inequalities. Then, this problem is reduced to (0,1)-CVP in the l-th norm and the given integer constant $r \neq 0$.

Problem F *Let*

$$f(Y) = \sqrt[l]{\sum_{i=1}^{m} |A_i \cdot Y - c_i|^l}$$

where $Y = [y_1, y_2, \ldots, y_n]$, $A_i \in \{0,1\}^n$ *and* $c_i \in \{0, r \neq 0\}$ *are given. Is there a vector* $Y \in Z^n$ *such that for a given constant* c,

$$f(Y) \leq c$$

Lemma 1 *Problem* F *is NP-complete.*

Proof It takes polynomial-time (in the problem size) to check whether $f(Y) \leq c$ for the given vector Y, thus the problem is in NP.

To reduce NOT-ALL-EQUAL problem to problem F we define first an integer variable with an odd value. Consider the given integer r of problem F. It can be represented as $2^q(2h+1)$, i.e. q is the highest power of two that is a divisor of r. Consider the sequence of variables $v_1^{(0)}, v_2^{(0)}, \ldots v_i^{(j)}, \ldots$, for $i = 1, 2, 3$ and $j = 1, \ldots, q$ satisfying the following inequalities

$$\begin{array}{ll} (j=0) & t_0 = |-r + v_1^{(0)}|^l + |v_1^{(0)} + v_2^{(0)}|^l \leq 0 \\ (j>0) & t_j = |v_2^{(j-1)} + v_1^{(j)} + v_3^{(j)}|^l + |v_1^{(j)} + v_2^{(j)}|^l + |v_3^{(j)} + v_2^{(j)}|^l \leq 0 \end{array} \quad (1)$$

that can be satisfied only if all terms are zeros. Thus, $v_1^{(0)} = -v_2^{(0)} = r$ and $v_1^{(j)} = -v_2^{(j)} = v_3^{(j)} = -v_2^{(j-1)}/2$, for $j > 0$. Hence, $v_2^{(q)} = -r/2^q = -2h - 1$. The above definition requires at most $3\lfloor \log_2 r \rfloor + 2$ variables (they define columns of the array A from problem F) and the same number of terms (that define rows of array A).

Let's consider an instance of NOT-ALL-EQUAL problem with a set $U = \{x_1, \ldots, x_k\}$ of k boolean variables and a collection $CL = \{cl_1, \ldots, cl_p\}$ of p clauses over U. For each boolean variable $x_j \in U$, we create a quadruplet $y_j, y_{j+k}, u_j, u_{j+k} \in Z$ of arithmetic variables. Each literal x_j is represented by y_j, whereas a literal $\bar{x_j}$ (i.e. boolean negation of x_j) is represented by y_{j+k}. Hence, each clause $cl_j = lt_{j1} \vee lt_{j2} \vee lt_{j3}$ in CL has the corresponding triplet of arithmetic variables y_{j1}, y_{j2}, y_{j3}. Consider inequality

$$s_1 = \sum_{i=1}^{2k} |y_i + u_i|^l + \sum_{i=1}^{k} |v_2^{(q)} + y_i + y_{i+k}|^l + \sum_{j=0}^{q} t_j \leq 0 \quad (2)$$

It is satisfiable (and becomes an equality) if and only if all terms yield zero, i.e. when $y_i = -u_i$ and $y_i + y_{\bar{i}} = 2h+1$ for all $i \leq 2k$ (from now on $y_{\bar{i}}$ will denote y_{i+k} if $i \leq k$, and y_{i-k} otherwise).

Inequality

$$s_2 = s_1 + \sum_{i=1}^{k} (s_1 + |y_i + u_{\bar{i}}|^l) \leq k \quad (3)$$

contains term s_1 repeated $k+1$ times. Hence, its solution satisfies $y_i = -u_i,\ y_i + y_{\bar{i}} = 2h+1$ and the following inequality also holds:

$$|y_i + u_{\bar{i}}|^l = |y_i - y_{\bar{i}}|^l = |2(y_i - h) - 1|^l \geq 1$$

Clearly, inequality (3) is satisfiable and becomes an equality if and only if

$$(\forall 1 \leq i \leq 2k : y_i = -u_i \wedge y_i \in \{h, h+1\}) \tag{4}$$

Finally, the truth assignment for U satisfies the NOT-ALL-EQUAL instance[1] if and only if the triplet of arithmetic variables corresponding to each clause $cl_j \in CL$ satisfies (4) and the following inequality (which, if holds, becomes an equality)

$$|y_{j1} + u_{\bar{j2}}|^l + |y_{j2} + u_{\bar{j3}}|^l + |y_{j3} + u_{\bar{j1}}|^l \leq 1 \tag{5}$$

The left hand side of inequality (5) is non-negative. For $u_i = -y_i,\ y_{\bar{i}} = 2h+1-y_i$, it is at least equal to "1". Hence, it is clear that the following inequality:

$$\sum_{i=1}^{p} s_1 + s_2 + \sum_{i=1}^{p}(|y_{i1} + u_{\bar{i2}}|^l + |y_{i2} + u_{\bar{i3}}|^l + |y_{i3} + u_{\bar{i1}}|^l) \leq p + k \tag{6}$$

can be satisfied if and only if there is a truth assignment solving the given NOT-ALL-EQUAL problem instance. Indeed, since term s_1 is repeated $p+k+1$ times on the left hand side of the inequality (6) then its solution satisfies (4), consequently the second term is no less than k and the third term no less than p. Hence, for properly selected matrix $A = [\ldots, A_i, \ldots] \in \{0,1\}^{n \times m}$, the following inequality can be satisfied if and only if there is a truth assignment solving the given NOT-ALL-EQUAL problem instance:

$$\sum_{i=1}^{m} |A_i \cdot Y - c_i|^l \leq p + k \tag{7}$$

where $Y = [v_1^{(0)}, v_2^{(0)}, \ldots v_3^{(q)}, \ldots, u_1, \ldots, u_{2k}, y_1, \ldots, y_{2k}] \in Z^n$, $c_i \in \{0, r\}$, $n \leq 4k + 3\lfloor \log_2 r \rfloor + 2$ and $m \leq 3(p+k+1)(k + \lfloor \log_2 r \rfloor + 1) + 2p - 1$.

Thus, for every instance of the NOT-ALL-EQUAL problem we can construct an instance of the problem F in polynomial time. Hence, problem F is NP-complete. □

A simple observation that knowing the minimum of a function $f(Y)$ is sufficient to decide whether the inequality $f(Y) \leq c$ holds for a given c leads to the following:

[1]It is possible to use 3SAT problem in reduction, but then the corresponding inequality would involve four additional variables $d_{j1} \ldots, d_{j4}$ for each each clause and would consists of eight terms:

$$|y_{j1} + d_{j1} + d_{j2}|^l + |y_{j2} + d_{j1} + d_{j2}|^l + |y_{j1} + u_{\bar{j2}}|^l + |y_{\bar{j3}} + d_{j1} + d_{j2}|^l + |y_{j3} + d_{j1} + d_{j3}|^l + \sum_{j=j1,j2,j3} |d_j + d_{j4}|^l \leq 2$$

Boolean variable x_j is then equivalent to a predicate (y_j mod 2).

Corollary *(0,1)-CVP in a finite norm l is NP-hard, i.e., the problem of finding a vector of integers $X = [x_1, x_2, \ldots, x_n]^T$ that minimizes*

$$\sum_{i=1}^{m} |A_i \cdot X - c_i|^l$$

for given $c_i \in \{0, r \neq 0\}$ and matrix $A = [A_1, \ldots, A_m] \in \{0,1\}^{n \times m}$ is NP-hard.

4 (0,1)-CVP Complexity in Infinity Norm

The complexity of the (0,1)-CVP in infinity norm depends on the elements of the input vector.

If all elements of the input vector are in $\{-1, 0, 1\}$ then (0,1)-CVP has a polynomial-time solution. First step of this solution is to use a polynomial-time algorithm for linear equation integer feasibility problem [1] to find the vector with a zero distance to the input vector or to show that such vector does not exist. In the latter case, the second step simply produces the zero vector as an answer. The zero vector is at distance "1" from the input vector and in infinity norm this is the smallest nonzero distance between the input vector and any vector in the lattice.

For all other input vectors there is an integer r, $|r| \geq 2$, such that elements of the input vector form the superset of $\{0, r\}$. The proof below covers a stronger case, when the input vector elements are exactly in $\{0, r\}$, $|r| \geq 2$. In the proof we will use an analogue of problem F defined as follows:

Problem F: *Let*

$$f(Y) = \max_{i=1,\ldots,m} |A_i \cdot Y - c_i|$$

where $Y = [y_1, y_2, \ldots, y_n]$, and $A_i \in \{0,1\}^n$, $c_i \in \{0, r\}$, where $|r| \geq 2$, are given. Is there a vector $Y \in Z^n$, such that for a given constant c,

$$f(Y) \leq c$$

The reduction of problem F to (0,1)-CVP in infinity norm is based on the same argument as the reduction of F to (0,1)-CVP in the l-th norm presented in the previous section. Thus, we will only show how the 3SAT problem can be reduced to problem F. We can assume, without loss of generality, that $r > 0$. If this is not the case, the proof given below can be recast with signs of all integer variables reversed (i.e. with boolean value *True* represented by "1"and not, as below, by "-1").

Let k denote the number of boolean variables in the given instance of 3SAT problem and p be the number of its clauses. For each boolean variable $x_j \in U$, a pair $y_j, y_{j+k} \in Z$ of arithmetic variables is created. By introducing inequalities over these arithmetic variables, we will ensure that a solution to inequalities exists if and only if there is a truth assignment satisfying

the considered 3SAT problem instance. A solution to the inequalities produces required truth assignment by setting a boolean variable x_j to predicate $y_j = -1$. Conversely, the truth assignment yields the solution to inequalities by setting $y_j = -1$, if x_j is *True* and $y_j = 0$ otherwise, and putting $y_{j+k} = -1 - y_j$. Following the definition of a distance in the infinity norm, left hand sides of introduced inequalities will consist of a max operator applied to absolute values of expressions. Each expression will contain a simple sum of variables and a constant "0" or "$-r$". The right hand side have to be an integer constant, the same for all introduced inequalities ("1" in our proof).

It is easy to verify that any solution to inequality

$$s_1 = \max_{i=1,\ldots,k} \{(|y_i|, |y_{\bar{i}}|, |y_i + y_{\bar{i}}|)\} \leq 1 \tag{8}$$

satisfies $y_i \in \{-1, 0, 1\}$, $y_i + y_{\bar{i}} \leq 1$, hence for any solution to the above inequality $y_i = -1$ implies that $y_{\bar{i}} > -1$.

Let $v_j^{(i)}$, where $j = 0, 1, 2$; $i = 0, \ldots, q$ and $q = \lfloor \log_2(r-1) \rfloor$ be a sequence of auxiliary variables defined as follows:

$$\begin{array}{ll} (i = 0) & \max_{j=0,1,2} |v_j^{(0)}| \leq 1 \\ (i > 0) & \max_{j=0,1,2} |v_j^{(i)} + \sum_{m=0}^{i-1} v_{(j+1) \bmod 2}^{(m)}| \leq 1 \end{array} \tag{9}$$

It can be shown by induction that the above inequalities restrict feasible solutions to $-2^i \leq v_j^{(i)} \leq 2^i$. Let b_i's, where $i = 0, \ldots, q$, denote digits of the binary representation of $r - 1$. The following inequality:

$$\max_{j=0,2} | - r + \sum_{\{i|b_i=1\}} v_j^{(i)}| \leq 1 \tag{10}$$

is satisfiable if and only if $v_0^{(i)} = v_2^{(i)} = 2^i$, for $i \leq q$, hence $v_0^{(0)} = v_2^{(0)} = 1$. It is easy to check that the 3SAT problem is satisfiable if and only if for each clause $cl_j = lt_{j1} \vee lt_{j2} \vee lt_{j3}$ there is such assignment to the corresponding arithmetic variables y_{j1}, y_{j2}, y_{j3} that they satisfy (8) and their sum is negative, or, equivalently, that they make the following inequality hold:

$$|v_0^{(0)} + v_2^{(0)} + y_{j_1} + y_{j_2} + y_{j_3}| \leq 1 \tag{11}$$

Hence, the following inequality is satisfied if and only if the corresponding 3SAT problem is satisfiable:

$$\max(s_1, \max_{i=1}^{p} |v_0^{(0)} + v_2^{(0)} + y_{j_1} + y_{j_2} + y_{j_3}|) \leq 1$$

Inequalities (9–11) can be rewritten in the matrix form as:

$$\max_i |A_i \cdot [v_0^{(0)}, \ldots, v_3^{(q)}, y_1, \ldots, y_{2k}] - c_i| \leq 1$$

where each row A_i is created from terms of inequalities (9 – 11), hence contains only $\{0, 1\}$ entires and there are only $3(k + \lfloor \log_2(r-1) \rfloor) + p + 5$ of such rows. Thus, we can reduce the given 3SAT problem instance to an instance of problem F in a polynomial time.

5 Conclusion

The paper establishes complexity of the well-known closest vector problem in an arbitrary norm in a lattice generated by a (0,1)-matrix. This result can be used to decide complexity of minimization problems in graphs and networks when the CVP lattice is generated by the graph or network incidence matrix with elements in $\{0, 1\}$ or $\{-1, 0, 1\}$. The authors used the presented result to establish complexity of the data alignment problem in SIMD architectures, an important compile-time optimization method for parallel processing [5].

Matrix A created during the presented reductions includes rows that are linearly dependent on others. The question then arises whether the complexity of CVP is influenced by the introduction of dependent rows, i.e. whether CVP is harder for lattices that are not full-dimensional. The negative answer can be obtained by using the Hermite normal form [2] to reduce CVP in a lattice generated by an arbitrary matrix to the problem in a lattice defined by a full dimensional matrix.

The presented complexity proofs use a reduction technique that can be useful in analyzing other minimization problems. The essence of this technique is to reduce a propositional logic problem to a satisfiability problem of an inequality over integers, which in turn is reduced to some minimization problem. The presented technique can also be applied to complexity analysis of the related *shortest vector problem* which is to find such nonzero vector in the given lattice that is closest to the origin. Using reductions similar to those presented in this paper, it is easy to show that this problem is NP-hard in infinity norm even if the lattice generating matrix elements are restricted to the set $\{0, 1, r\}$, where $|r| > 1$. Similarly as for the corresponding (0,1)-CVP, the shortest vector problem in infinity norm has a (trivial) polynomial-time solution if the lattice generating matrix has all its elements in $\{-1, 0, 1\}$.

References

[1] R. Kannan, "Minkowski's Convex Body Theorem and Integer Programming," *Math. Operations Res.* Vol. 12, No. 3, August 1987, pp. 415-440

[2] R. Kannan and A. Bachem, "Polynomial Time Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix", *SIAM J. Comput.* Vol. 8, 1979, pp. 499-507.

[3] R. M. Karp, "Reducibility among Combinatorial Problems" in R. E. Miller and J. W. Thatcher (Eds.), *Complexity of Computer Computations.*, Plenum Press, New York, 1972, pp. 85-103.

[4] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, John-Wiley and Sons, 1988.

[5] B. Sinharoy and B. K. Szymanski, "Data Alignment in SIMD Machines," submitted to *IEEE Trans. on Parallel and Distributed Systems*, also Technical Report 91-10, Department of Computer Science, RPI, May, 1991.

[6] P. van Emde Boas, *Another NP-complete Problem and the Complexity of Computing Short Vectors in a Lattice*, Rep. 81-04, Math. Inst. Univ. Amsterdam, 1981.