Linear Transformation in Pseudo-Boolean Functions

Yong-Hyuk Kim
Department of Computer Science and Engineering
Kwangwoon University
Wolgye-dong, Nowon-gu, Seoul, 139-701, Korea
yhdfly@kw.ac.kr

Categories and Subject Descriptors

G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra—*Matrix Inversion*; G.2.3 [**Discrete Mathematics**]: Applications

General Terms

Experimentation, Theory

Keywords

Pseudo-boolean function, coordinate-change, nonsingular binary matrix

1. INTRODUCTION

Traditional approaches dealing with pseudo-boolean function mostly use the inherent standard basis. When we consider another basis instead of the standard one, the linkage structure between basis elements and the ruggedness of the problem space can be completely different from original ones. Through a change of basis, a complex/difficult problem may be changed into a simple/easy one and vice versa.

Chryssomalakos and Stephens [4] theoretically dealt with the bases of function space on $\mathbb{Z}_2^n$. In this paper, we investigate bases of more fundamental space, i.e., those of the vector space $\mathbb{Z}_2^n = (\{0,1\}^n, \oplus)$.[1]

Gene reordering [3, 8] can be considered as a special case of the change of basis. If T is just a permutation matrix, a change of basis means a reordering of gene positions in encoding. The concept of changing basis is much more general than that of gene reordering.

Coordinate changes based on eigenspace and orthogonalization have been studied [10, 11]. But, they focused on real-code representation. These results cannot apply to pseudo-boolean functions using binary representation because of the following proposition.

PROPOSITION 1. *Let T be a nonsingular binary matrix. Then, if T is not the identity matrix I, the eigenspace of T does not span $\mathbb{Z}_2^n$.*

PROOF. The spectrum of T is a subset of $\mathbb{Z}_2$. Since T is nonsingular, zero cannot be an eigenvalue of T. If T has an eigenvalue, it must be one. Suppose that the eigenspace of T spans $\mathbb{Z}_2^n$. Then there exist linearly-independent n eigenvectors with one as eigenvalue. Let A be the matrix in columns of which has such n eigenvectors. Then $TA = A$. Since A is nonsingular, $T = I$. This is a contradiction. □

[1] $\oplus$ is the exclusive-or (XOR) operator. $(a_1, a_2, \ldots, a_n) \oplus (b_1, b_2, \ldots, b_n) = (a_1 \oplus b_1, a_2 \oplus b_2, \ldots, a_n \oplus b_n)$.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

2. COORDINATE-CHANGE

A matrix A is called *binary* if $A \in M_{n\times n}(\mathbb{Z}_2)$. Binary matrices have been widely used to deal with the adjacency of a graph [1, 2]. In particular, Anderson and Feil [1] transformed the *light bulb puzzle* into the problem of solving a linear system $A\boldsymbol{x} = \boldsymbol{b}$ from its graph structure, where $A \in M_{n\times n}(\mathbb{Z}_2)$ and $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{Z}_2^n$. Then the solution could be obtained by computing the inverse of A, i.e., $\boldsymbol{x} = A^{-1}\boldsymbol{b}$. Binary matrices are also useful in dealing with the *cut/cycle* subspace of a graph, which is a vector space over $\mathbb{Z}_2$ [2, 5]. They can also be used to represent a change of basis of a vector space over $\mathbb{Z}_2$ in combinatorial problems. In this paper, we will present this new application of binary matrices.

A *basis* for a vector space of dimension n is a sequence of n vectors with the property that every vector in the space can be uniquely expressed as a linear combination of the basis vectors. Since it is often desirable to work with more than one basis for a vector space, it is important to be able to easily transform coordinate-wise representations of vectors and linear transformations taken with respect to one basis to their equivalent representations with respect to another basis. Such a transformation is called a *change of basis.* The following theorem is easily induced from the basic theory of linear algebra [6].

THEOREM 1. *Let $\mathfrak{B}_1$ and $\mathfrak{B}_2$ be two bases for $\mathbb{Z}_2^n$. Then there exists a nonsingular matrix $T \in M_{n\times n}(\mathbb{Z}_2)$ such that for every $v \in \mathbb{Z}_2^n$, $T(v)_{\mathfrak{B}_1} = (v)_{\mathfrak{B}_2}$, where $(v)_{\mathfrak{B}}$ is the representation of v with respect to the basis $\mathfrak{B}$.*

The matrix T in Theorem 1 is called *coordinate-change matrix* from the basis $\mathfrak{B}_1$ to $\mathfrak{B}_2$.

The standard basis $\mathfrak{B}_s$ for $\mathbb{Z}_2^n$ is $\{e_1, e_2, \ldots, e_n\}$, where $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the element of $\mathbb{Z}_2^n$ with 1 only in the j-th place and 0s elsewhere. Given a nonsingular matrix $T \in M_{n\times n}(\mathbb{Z}_2)$, the matrix T can be regarded as coordinate-change matrix from the standard basis to another basis $\mathfrak{B}_T$ related to T. That is, for every $v \in \mathbb{Z}_2^n$, $T(v)_{\mathfrak{B}_s} = (v)_{\mathfrak{B}_T}$ and then $\mathfrak{B}_T = \{Te_1, Te_2, \ldots, Te_n\}$. Hence the problem of finding a new basis becomes that of finding a proper nonsingular binary matrix.

3. EXPERIMENTS

We tested on two n-dimensional pseudo-boolean functions. These functions are simplified adaptations of Kauffman's NK landscapes [7]. Kauffman defined a function with n bits, in which each bit's fitness contribution depends on its k neighbors. NK landscapes thus have "tunable ruggedness" and are often used to test genetic algorithms. We defined two functions with $k = 1, n-2$, respectively. Each variable's fitness contribution depends on its next k variables. Three test functions are given in the following.

$$F_2(\boldsymbol{x}) = \sum_{i=1}^{n-1} x_i \oplus x_{i+1} + x_n \oplus x_1 \text{ and}$$

$$F_{n-1}(\boldsymbol{x}) = \sum_{i=1}^{n} x_1 \oplus \cdots \oplus x_{i-1} \oplus x_{i+1} \cdots \oplus x_n,$$

where each $x_i \in \mathbb{Z}_2$.

Suppose that T is a nonsingular binary matrix. Let $\boldsymbol{x} = (x_1\ x_2\ \cdots\ x_n)^T$ be $(v)_{\mathfrak{B}_s}$ and $\boldsymbol{y} = (y_1\ y_2\ \cdots\ y_n)^T$ be $(v)_{\mathfrak{B}_T}$, where $v \in \mathbb{Z}_2^n$. Then $T\boldsymbol{x} = \boldsymbol{y}$. Hence, $F_2(\boldsymbol{x}) = F_2(T^{-1}\boldsymbol{y})$ and $F_{n-1}(\boldsymbol{x}) = F_{n-1}(T^{-1}\boldsymbol{y})$.

For the function F_2, we choose the coordinate-change matrix $T = (t_{i,j})$ such that $t_{i,i} = t_{i,i+1} = 1$ and $t_{i,j} = 0$ elsewhere. For the function F_{n-1}, we choose the matrix $T = (t_{i,j})$ such that $t_{i,i} = 0$ and $t_{i,j} = 1$ elsewhere. Then T is nonsingular when n is even, because $T^2 = (\mathbf{1} - I)^2 = O - 2 \cdot \mathbf{1} + I = I$, where $\mathbf{1}$ is an $n \times n$ binary matrix of which each element is always 1. Given a coordinate-change matrix T, its inverse T^{-1} can be efficiently computed by the Gaussian elimination method. In the case of $n = 6$,

$$\begin{pmatrix} 1&1&0&0&0&0 \\ 0&1&1&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&1&1 \\ 0&0&0&0&0&1 \end{pmatrix} \quad \begin{pmatrix} 1&1&1&1&1&1 \\ 0&1&1&1&1&1 \\ 0&0&1&1&1&1 \\ 0&0&0&1&1&1 \\ 0&0&0&0&1&1 \\ 0&0&0&0&0&1 \end{pmatrix} \quad \begin{pmatrix} 0&1&1&1&1&1 \\ 1&0&1&1&1&1 \\ 1&1&0&1&1&1 \\ 1&1&1&0&1&1 \\ 1&1&1&1&0&1 \\ 1&1&1&1&1&0 \end{pmatrix} \quad \begin{pmatrix} 0&1&1&1&1&1 \\ 1&0&1&1&1&1 \\ 1&1&0&1&1&1 \\ 1&1&1&0&1&1 \\ 1&1&1&1&0&1 \\ 1&1&1&1&1&0 \end{pmatrix}$$

T for F_2 $\quad$ T^{-1} for F_2 $\quad$ T for F_{n-1} $\quad$ T^{-1} for F_{n-1}.

Hence we can rewrite the functions F_2 and F_{n-1} based the new basis $\mathfrak{B}_T$ as follows:

$$F_2(T^{-1}\boldsymbol{y}) = \sum_{i=1}^{n-1} y_i + \bigoplus_{i=1}^{n-1} y_i \text{ and}$$

$$F_{n-1}(T^{-1}\boldsymbol{y}) = \sum_{i=1}^{n} y_i,$$

where each $y_i \in \mathbb{Z}_2$.

For test, we used the genetic algorithm of [9] with typical parameter setting. The genetic algorithm maximizes test functions. All genetic parameters but evaluation function are the same and given in the following.

number of variables (n)	40
population size	100
selection	tournament selection with size 2
crossover rate crossover type	1.0 one-point crossover
mutation rate mutation type	0.1 gene-wise mutation
replacement proportion	0.75
maximum number of generations	200

Table 1 shows the results under the same genetic framework. The optimal value of each test function is 40. Each figure of the table came from 30 runs. For every case, after changing the standard basis into another nice one, the performance was significantly improved. In particular, test on the function F_{n-1} shows that the new basis is much better than the standard basis. It naturally comes from the fact that the new basis completely removes the strong linkage structure inherent in the function represented by the standard basis.

Table 1: Results ($n = 40$)

Test Function	Best	Ave (%-gap)	Std	Optimal Value
$F_2(\boldsymbol{x})$	**40**	38.27 (4.33)	0.87	40
$F_2(T^{-1}\boldsymbol{y})$	**40**	**39.27 (1.83)**	0.98	40
$F_{n-1}(\boldsymbol{x})$	33	29.37 (26.58)	1.38	40
$F_{n-1}(T^{-1}\boldsymbol{y})$	**40**	**39.73 (0.68)**	0.45	40

From 30 trials.
The value of %-gap is computed as follows:
$100 \times \frac{\text{(optimum}-\text{average)}}{\text{optimum}}$.

However, this paper does not give any hints about what basis should be chosen to make the search space smooth. More studies about the mechanism to find a good basis, i.e., a good coordinate-change matrix, would be promising. To do this, it will be a good initial study to investigate the space of nonsingular binary matrices, i.e., general linear group over $\mathbb{Z}_2^n$. Some problems may hardly be affected by a change of basis. Research about what problems to be affected by changing basis and how much for the problems to be affected is also left for future study.

4. REFERENCES

[1] Anderson, M., and Feil, T. Turning lights out with linear algebra. *Mathematics Magazine 71*, 4 (1998), 300–303.

[2] Biggs, N. *Algebraic Graph Theory*, second ed. Cambridge University Press, 1994.

[3] Bui, T. N., and Moon, B. R. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers 45*, 7 (1996), 841–855.

[4] Chryssomalakos, C., and Stephens, C. R. What basis for genetic dynamics? In *Proceedings of the Genetic and Evolutionary Computation Conference* (2004), pp. 1018–1029.

[5] Diestel, R. *Graph Theory*, third ed. Springer-Verlag, Heidelberg, 2005. Graduate Texts in Mathematics, Volume 173.

[6] Friedberg, S. H., Insel, A. J., and Spence, L. E. *Linear Algebra*, third ed. Prentice-Hall International, Inc., 1997.

[7] Kauffman, S. Adaptation on rugged fitness landscapes. *Lectures in the Science of Complexity* (1989), 527–618.

[8] Kim, Y.-H., Kwon, Y.-K., and Moon, B.-R. Problem-independent schema synthesis for genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2003), pp. 1112–1122.

[9] Sastry, K. Single and multiobjective genetic algorithm toolbox in C++. Tech. Rep. 2007016, IlliGAL, University of Illinois at Urbana-Champaign, June 2007.

[10] Whitley, D., Lunacek, M., and Knight, J. Ruffled by ridges: How evolutionary algorithms can fail. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2004), pp. 294–306.

[11] Wyatt, D., and Lipson, H. Finding building blocks through eigenstructure adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference* (2003), pp. 1518–1529.