A method for traveling salesman problem by use of pattern processing with image compression

Kouichi Nitta, Shinichi Ohta, and Osamu Matoba

Department of Computer Science and Systems Engineering, Graduate School of Engineering, Kobe University, Address: Rokkodai 1-1, Nada, Kobe 657-8501, Japan;

ABSTRACT

In this research, a scheme for SIMD (Single Instruction stream Multi Data stream) pattern processing for two dimensional (2D) image data is applied to the traveling salesman problem (TSP). Firstly, 2D SIMD pattern processing for the TSP is designed. In the processing, two kinds of image data are prepared to represent graph data. By cross correlation between the prepared images, a set of pathlength is obtained. The tour with the minimum path length is extracted from post pattern processing. Numerical analysis verifies that the scheme is effective for the TSP.

Keywords: spatial coding, SIMD pattern operation, large scale information processing, image compression, traveling salesman problem

1. INTRODUCTION

Some schemes for ultra large scale processing are much attractive for various problems with computational hard. Quantum computing and DNA computing are mentioned as examples of them. In such a situation, we research on SIMD (Single Instruction Multiple Data) type two dimensional (2D) pattern processing. This processing has inherently been proposed in digital optical computing.[1] It is considered to be useful for not only optical implementation but also electronic one.

We have developed an effective scheme based on the processing. In this scheme, image compression and operations for compressed datum are utilized for large scale processing. We have applied the scheme to a solution for prime factorization. It has been verified that the solution is much effective in prime factorization.

In this research, we apply the scheme to the traveling salesman problem (TSP). TSP is one of the NP complete problems and some methods for this problem has been reported the research filed of optical signal processing.[2,3] Especially, this research is inspired by the solution presented in Ref.[3]

In our solution, firstly, 2D SIMD pattern processing for the TSP is designed. In the processing, two kinds of image data are prepared as inputs. One shows a set of path for touring. Information on one path is coded a row vector. Elements in a vector data are binary formats. A set of vectors are aligned in column direction and converted to a 2D discrete image to represent all touring. The other also indicates distance between nodes. A bit pattern of distance between two nodes is coded as row datum. The datum is converted binary row patterns. Then, whole sets of the binary patterns are aligned in column direction. As a result, a 2D binary image is prepared. By cross correlation between the prepared images, a set of pathlength is obtained. The tour with the minimum path length is extracted from post pattern processing.

Section 2 summarizes a scheme for SIMD pattern processing with image compression. Section 3 describes a principle to solve the TSP. Section 4 shows a proposed solution of the TSP and estimates the solution.

K.N.: E-mail: nitta@kobe-u.ac.jp, FAX: +81(78) 803 6390
O.M.: E-mail: matoba@kobe-u.ac.jp, FAX: +81 (78) 803 6390

Optics and Photonics for Information Processing III, edited by Khan M. Iftekharuddin, Abdul Ahad Sami Awwal,
Proc. of SPIE Vol. 7442, 74420B · © 2009 SPIE · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.826709

2. SIMD PATTERN PROCESSING WITH IMAGE COMPRESSION

2.1. 2D SIMD processing

2D SIMD pattern processing has inherently been studied in digital optical computing. Recently, we have proposed a scheme based on the pattern processing. In the scheme, massive data processing equivalent to 2D pattern operations in accordance with optical array logic (OAL). OAL is one of the paradigm for digital optical computing.[1]

In OAL, input and output are given as 2D discrete images. Pixels on these images are binary formats. Let us consider that $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ show the values at (i, j) pixels on the two input and the output images, the contents of operations are represented as Eq. (1).

$$c_{i,j} = \sum_{m=-L}^{L} \sum_{n=-L}^{L} f_{m,n}(a_{i+m,j+n}, b_{i+m,j+n}) \qquad (i, j = 1, ..., N) \tag{1}$$

In Eq. (1), $f_{m,n}$ denotes a content of the operation. A basic format of $f_{m,n}$ is given as a logical operation between neighborhood pixels. Figure 1 describes typical examples of basic operations. (a), (b), and (c) in this figure depict NOT, AND, XOR operations, respectively. Also, (d) and (e) in the figure shows pattern shift and pattern expansion. These are examples of operations with neighborhood pixels.

A sequence of these basic operations can treat some solutions of parallel programing. Recently, we have studied on a procedure for prime factorization.[4] In the procedure, image size required for prime factorization grows exponentially in bit length of a target integer. To solve such a problem, we have proposed a effective scheme for SIMD pattern processing. In this scheme, some operations equivalent to parallel programming of 2D SIMD processing is executed with less processing costs by use of image compression.

A brief procedure of the proposed scheme is shown in Fig. 2. A method for data compression is applied to input images. With this operation, information on an input image is represented as a set of datum. Sets of input datum are operated by a sequence of elemental gates, and output datum are obtained. Finally output images are given by the decoding process.

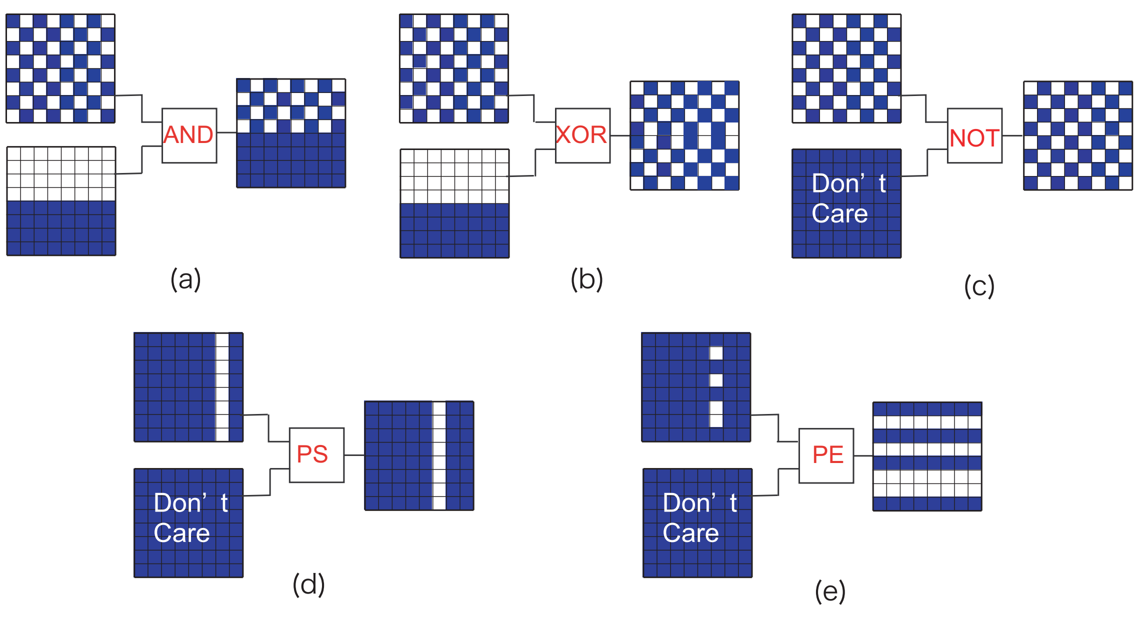

Figure 1. Basic operations in 2D pattern processing: (a) AND, (b) XOR, (c) NOT operations, (d) pattern shift, and (e) pattern expansion.

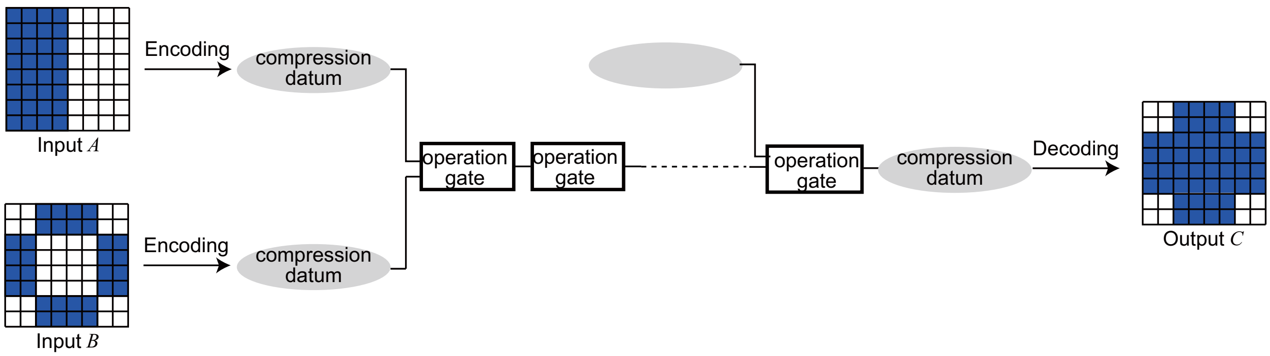

Figure 2. Diagram of the basic concept of SIMD pattern processing

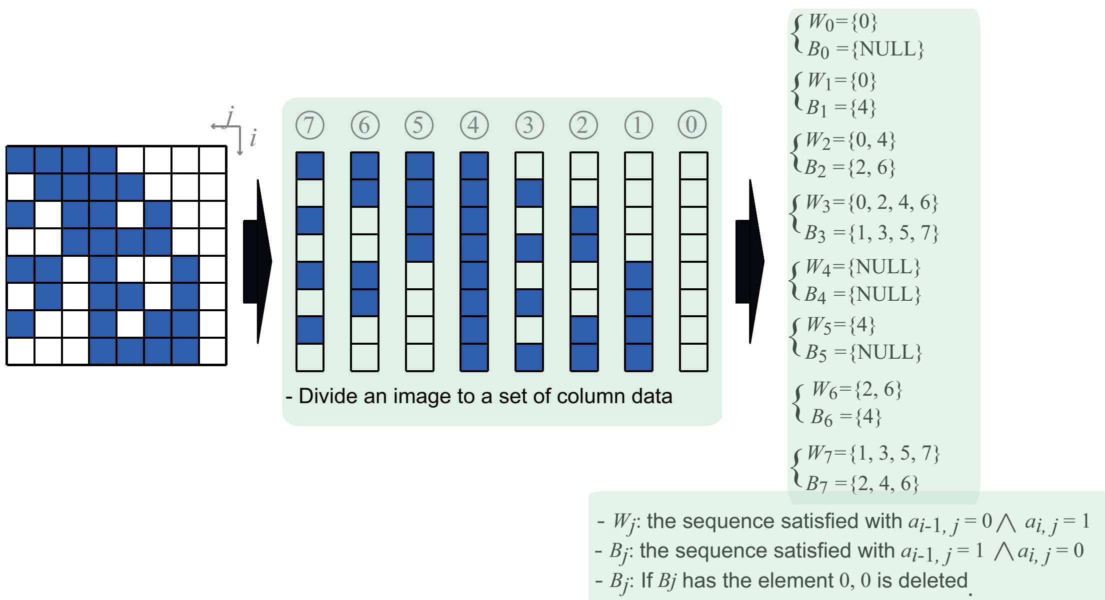

Figure 3. Schematic diagram of the encoding.

This scheme requires a lossless compression because output images should be completely reconstructed as final results. Moreover, elemental gates equivalent to operations as described in Fig .1 should be constructed to implement the processing designed as parallel programming in 2D pattern processing.

2.2. A encoding (decoding) and basic operations for compressed datum

A method for the proposed scheme have been developed in our previous work.[5] This method is applied to an algorithm for prime factorization. We also employ this method for the TSP. The procedure of the method is explained as following.

Figure 3 shows an example to generate a coded datum. 2D patterns in an input image are divided into a set of column data before encoding. Two kinds of datum defined as W_j and B_j are prepared in the data compression. W_j consists of a set of $i's$ satisfied with Eq. (2).

$$a_{i-1,j} = 0 \cap a_{i,j} = 1 \tag{2}$$

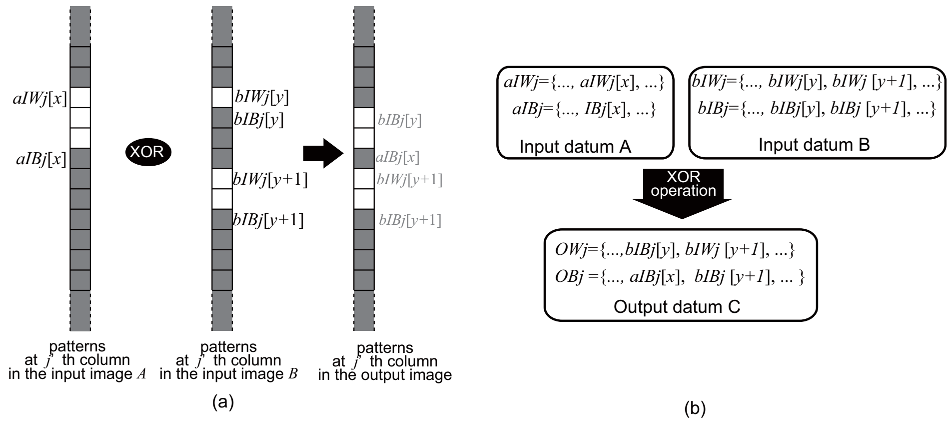

Figure 4. Schematic diagram of a XOR gate: (a) a relation between input and output patterns, (b) the relation between corresponding compression datum.

Here, the element '0' is added at the first element in the W_j if $a_{0,j}$ is '0'. On the other hand, the contents of B_j are given as a set of i's satisfied with Eq. (3).

$$a_{i-1,j} = 1 \cap a_{i,j} = 0 \tag{3}$$

We have presented NOT, AND, and XOR gates. The principle of the XOR operation is explained. In the simple pattern processing, this operation is defined as

$$c_{i,j} = \begin{cases} 0 & \text{if} \quad a_{i,j} = b_{i,j} \\ 1 & \text{other.} \end{cases} \tag{4}$$

A relation between parts of input images and that of the output one is shown in Fig. 4. (a) and (b) in this figure indicate data flows of the gate in both image and coression domains, respectively. For simplicity, only patterns of one coloumn are extracted from 2D images. From Fig. 4, output patterns are inverted at an i'th row if i is satisfied with the following equation.

$$((a_{i,j-1} \neq a_{i,j}) \cap (b_{i,j-1} = b_{i,j})) \cup ((a_{i,j-1} = a_{i,j}) \cap (b_{i,j-1} \neq b_{i,j})) \tag{5}$$

Figure 5 shows a procedure of the XOR gate for compressed datum. First process is generation of a sequence from input datum. This sequence includes all elements in the input datum. Moreover, these elements are sorted in the sequence. Next, if there are the same elements in the sequence, such elements are deleted. This is because such a set of the elements shows that both input images are inverted at the same pixels. Third process, as described in Fig. 5, odd elements in the sequence and even ones are output as OW_j and OB_j, respectively. Procedures for NOT and AND gates are explained in Ref.[5]

3. TRAVELLING SALESMAN PROBLEM

Fgiure 6 shows an example of a graph for the TSP. This example consists of four nodes and the weight values between cities are asymmetric. The purpose of the TSP is to derive the shortest tour through all nodes. Considering that the number of nodes is N, $(N-1)!$ tours are given as candidates of the desired path. Therefore, 6 tours should be investigated as described in Eq. (6).

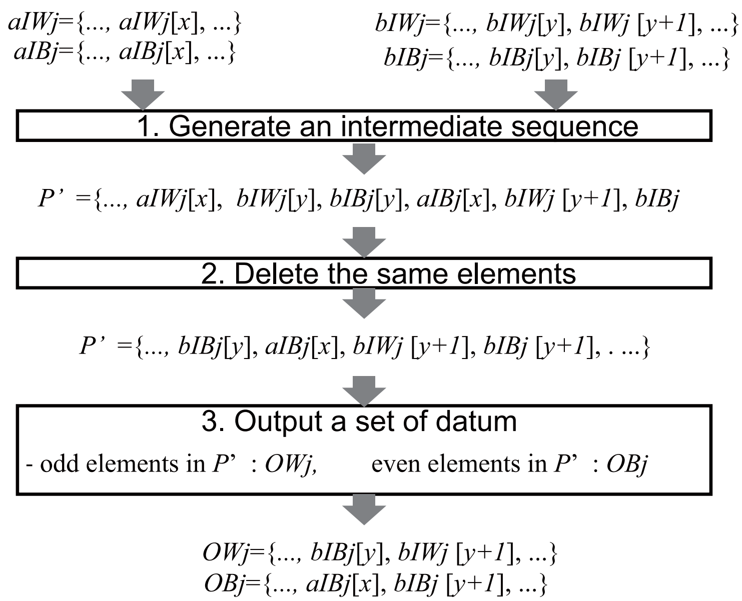

Figure 5. A procedure for an XOR gate.

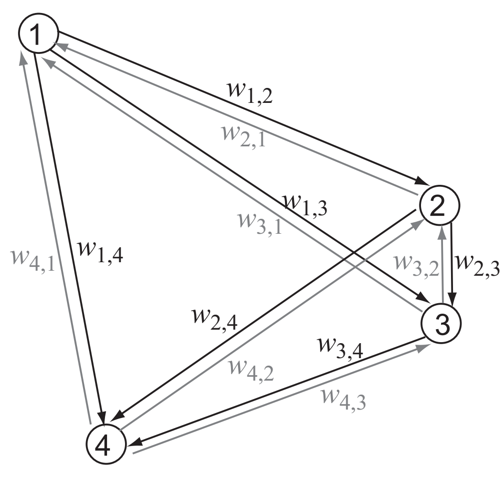

Figure 6. An example of graph in the TSP at $N = 4$.

$$
\begin{aligned}
&T1: \text{node1} \rightarrow \text{node2} \rightarrow \text{node3} \rightarrow \text{node4} \rightarrow \text{node1}\\
&T2: \text{node1} \rightarrow \text{node3} \rightarrow \text{node4} \rightarrow \text{node2} \rightarrow \text{node1}\\
&T3: \text{node1} \rightarrow \text{node4} \rightarrow \text{node2} \rightarrow \text{node3} \rightarrow \text{node1}\\
&T4: \text{node1} \rightarrow \text{node4} \rightarrow \text{node3} \rightarrow \text{node2} \rightarrow \text{node1}\\
&T5: \text{node1} \rightarrow \text{node2} \rightarrow \text{node4} \rightarrow \text{node3} \rightarrow \text{node1}\\
&T6: \text{node1} \rightarrow \text{node3} \rightarrow \text{node2} \rightarrow \text{node4} \rightarrow \text{node1}
\end{aligned} \tag{6}
$$

Here, length of these tours are represented as Eq. (7).

$$
\begin{aligned}
T1: l_1 &= w_{1,2} + w_{2,3} + w_{3,4} + w_{4,1}\\
T2: l_2 &= w_{1,3} + w_{3,4} + w_{4,2} + w_{2,1}\\
T3: l_3 &= w_{1,4} + w_{4,2} + w_{2,3} + w_{3,1}\\
T4: l_4 &= w_{1,4} + w_{4,3} + w_{3,2} + w_{2,1}\\
T5: l_5 &= w_{1,2} + w_{2,4} + w_{4,3} + w_{3,1}\\
T6: l_6 &= w_{1,3} + w_{3,2} + w_{2,4} + w_{4,1}
\end{aligned} \tag{7}
$$

In the above equation, $w_{i,j}$ shows weight vaule from the node i to the node j.

The solution presented in Ref.,[3] a mathematical procedure is reported. The weight vectors are defines as Eq. (8).

$$
\mathbf{w} = [w_{1,2} \quad w_{1,3} \quad w_{1,4} \quad w_{2,1} \quad w_{2,3} \quad w_{2,4} \quad w_{3,2} \quad w_{3,1} \quad w_{3,4} \quad w_{4,2} \quad w_{4,3} \quad w_{4,1}]^{\mathbf{T}} \tag{8}
$$

Informations on the tours shown in Eq. (6) is also represented as a 2D matrix $\mathbf{b}$ as shown in Eq. (9).

$$
\mathbf{b} = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1\\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0\\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1\\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \tag{9}
$$

From Eqs. (8) and (9), a set of length of all tours is derived with Eq. (10).

$$
\begin{aligned}
\mathbf{l} = \mathbf{b} \cdot \mathbf{w} &= \begin{bmatrix}
w_{1,2} + 0 + 0 + 0 + w_{2,3} + 0 + 0 + 0 + w_{3,4} + 0 + 0 + w_{4,1}\\
w_{1,2} + 0 + 0 + 0 + 0 + w_{2,4} + 0 + w_{3,1} + 0 + 0 + w_{4,3} + 0\\
0 + w_{1,3} + 0 + 0 + 0 + w_{2,4} + w_{3,2} + 0 + 0 + 0 + 0 + w_{4,1}\\
0 + w_{1,3} + 0 + w_{2,1} + 0 + 0 + 0 + 0 + w_{3,4} + w_{4,2} + 0 + 0\\
0 + 0 + w_{1,4} + 0 + w_{2,3} + 0 + 0 + w_{3,1} + 0 + w_{4,2} + 0 + 0\\
0 + 0 + w_{1,4} + w_{2,1} + 0 + 0 + w_{3,2} + 0 + 0 + 0 + w_{4,3} + 0
\end{bmatrix}\\
&= \begin{bmatrix}
w_{1,2} + w_{2,3} + w_{3,4} + w_{4,1}\\
w_{1,2} + w_{2,4} + w_{3,1} + w_{4,3}\\
w_{1,3} + w_{2,4} + w_{3,2} + w_{4,1}\\
w_{1,3} + w_{2,1} + w_{3,4} + w_{4,2}\\
w_{1,4} + w_{2,3} + w_{3,1} + w_{4,2}\\
w_{1,4} + w_{2,1} + w_{3,2} + w_{4,3}
\end{bmatrix} = \begin{bmatrix}
l_1\\ l_2\\ l_3\\ l_4\\ l_5\\ l_6
\end{bmatrix}
\end{aligned} \tag{10}
$$

4. A SOLUTION FOR THE TSP WITH 2D SIMD PATTERN PROCESSING

In this section, we show the proposed method for the TSP. At first, two kinds of images are prepared for input data corresponding to the matrix $\mathbf{b}$ and the vector $\mathbf{w}$. Figure 4 shows a schematic diagram of the image

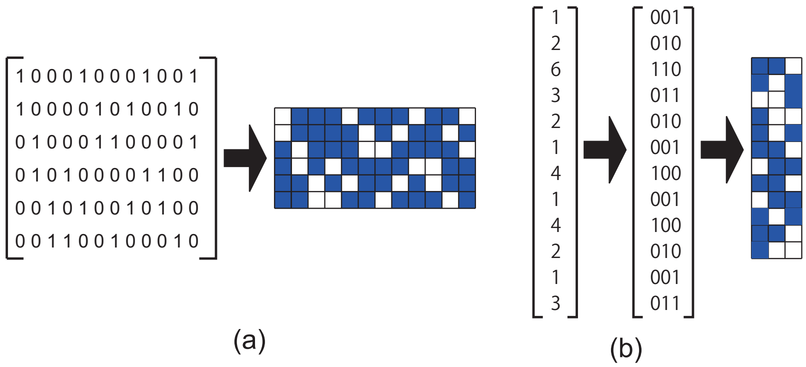

Figure 7. Image representations for the matrix **b** (a) and the weight vectors **w** (b).

representations. Conversion for the matrix **b** is simple. An (m, n) element in the matrix **b** is coded at (m, n) pixel in one input image. In the image representation for **w**, bit patterns of the weight values are considered. Here Eq. (11) shows a generalized form of matrix operations for the TSP.

$$
\begin{aligned}
\mathbf{l} = \mathbf{b} \cdot \mathbf{w} &= \begin{pmatrix}
b_{0,0} & b_{1,0} & b_{2,0} & b_{3,0} & b_{4,0} & b_{5,0} & b_{6,0} & b_{7,0} & b_{8,0} & b_{9,0} & b_{10,0} & b_{11,0} \\
b_{0,1} & b_{1,1} & b_{2,1} & b_{3,1} & b_{4,1} & b_{5,1} & b_{6,1} & b_{7,1} & b_{8,1} & b_{9,1} & b_{10,1} & b_{11,1} \\
b_{0,2} & b_{1,2} & b_{2,2} & b_{3,2} & b_{4,2} & b_{5,2} & b_{6,2} & b_{7,2} & b_{8,2} & b_{9,2} & b_{10,2} & b_{11,2} \\
b_{0,3} & b_{1,3} & b_{2,3} & b_{3,3} & b_{4,3} & b_{5,3} & b_{6,3} & b_{7,3} & b_{8,3} & b_{9,3} & b_{10,3} & b_{11,3} \\
b_{0,4} & b_{1,4} & b_{2,4} & b_{3,4} & b_{4,4} & b_{5,4} & b_{6,4} & b_{7,4} & b_{8,4} & b_{9,4} & b_{10,4} & b_{11,4} \\
b_{0,5} & b_{1,5} & b_{2,5} & b_{3,5} & b_{4,5} & b_{5,5} & b_{6,5} & b_{7,5} & b_{8,5} & b_{9,5} & b_{10,5} & b_{11,5}
\end{pmatrix} \cdot \begin{pmatrix}
w_{1,2} \\ w_{1,3} \\ w_{1,4} \\ w_{2,1} \\ w_{2,3} \\ w_{2,4} \\ w_{3,2} \\ w_{3,1} \\ w_{3,4} \\ w_{4,2} \\ w_{4,3} \\ w_{4,1}
\end{pmatrix} \\
&= \begin{pmatrix}
b_{0,0}w_{1,2} + b_{1,0}w_{1,3} + \ldots + b_{11,0}w_{4,1} \\
b_{0,1}w_{1,2} + b_{1,1}w_{1,3} + \ldots + b_{11,1}w_{4,1} \\
b_{0,2}w_{1,2} + b_{1,2}w_{1,3} + \ldots + b_{11,2}w_{4,1} \\
b_{0,3}w_{1,2} + b_{1,3}w_{1,3} + \ldots + b_{11,3}w_{4,1} \\
b_{0,4}w_{1,2} + b_{1,4}w_{1,3} + \ldots + b_{11,4}w_{4,1} \\
b_{0,5}w_{1,2} + b_{1,5}w_{1,3} + \ldots + b_{11,5}w_{4,1}
\end{pmatrix} \\
&= \begin{pmatrix}
A_{0,0} + A_{1,0} + \ldots + A_{11,0} \\
A_{0,1} + A_{1,1} + \ldots + A_{11,1} \\
A_{0,2} + A_{1,2} + \ldots + A_{11,2} \\
A_{0,3} + A_{1,3} + \ldots + A_{11,3} \\
A_{0,4} + A_{1,4} + \ldots + A_{11,4} \\
A_{0,5} + A_{1,5} + \ldots + A_{11,5}
\end{pmatrix} \\
&= \begin{pmatrix}
\sum_{i=0}^{(N-1)!-1} A_{i,0} \\
\sum_{i=0}^{(N-1)!-1} A_{i,1} \\
\sum_{i=0}^{(N-1)!-1} A_{i,2} \\
\sum_{i=0}^{(N-1)!-1} A_{i,3} \\
\sum_{i=0}^{(N-1)!-1} A_{i,4} \\
\sum_{i=0}^{(N-1)!-1} A_{i,5}
\end{pmatrix}
\end{aligned}
\tag{11}
$$

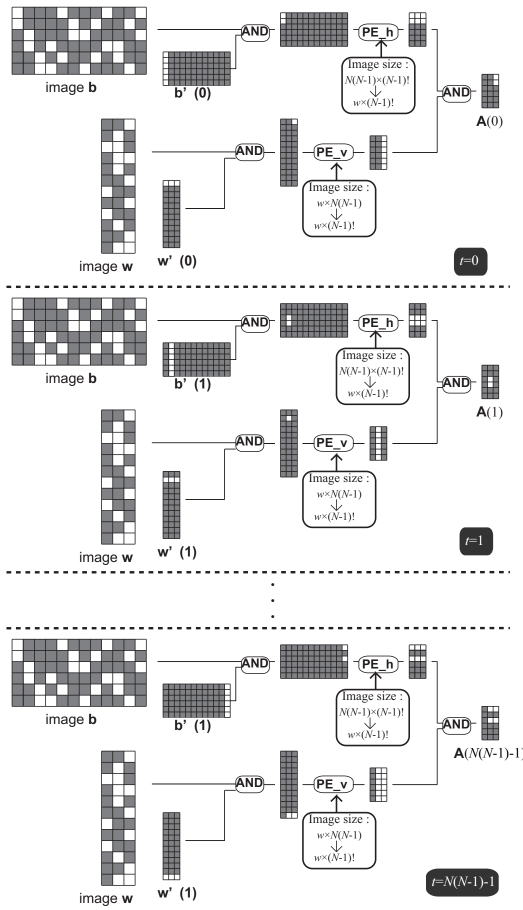

Figure 8. Block diagram for derivation of $A_{i,j}$.

In this eqaution, $A_{i,j}$ shows $b_{i,j} \times w'_i$ where w'_i is an ith element in $\mathbf{w}$. In our method, a set of $A_{i,j}$ is derived. Figure 4 describes a flow of the derivations process. At $t = T$, $A_{T,j}$ is obtained from $j = 0$ to $(N-1)!$. Horizontal patterns of an ith column in $A(T)$ correspond to $A_{T,j}$ shown in Eq. (11). Operations for summations described in Eq. (11) are executed with the sequence for addition presented in our previous work.[5] Finally, the tour with the minimum costs is extracted with specific 2D pattern operations.

We implement the proposed method on a personal computer and verify the method of network models with small nodes. Figure 4 describes the relations between processing time and the number of nodes. Simple 2D pattern processing which does not use image compression is also estimated to compare with the proposed solution. As results of comparison, it is confirmed that proposed solution is useful for the TSP. From Fig. 4, the larger the number of node is, the more effective the solution is. However, the processing time grow exponentially in the solution. We should modify the procedure to make the solution more effective.

5. SUMMARY

We have proposed a solution for the TSP. The solution is based on a scheme for SIMD pattern processing with image compression. Note that this research is inspired by an optical method reported by N. T. Shaked.[3] Image representation for graph data in the the TSP is constructed. In the representation, two kinds of images are generated. One image is to describe all tours and the other is for weight vectors between nodes. We have designed a procedure of 2D pattern process and the scheme with image compression is applied to the procedure.

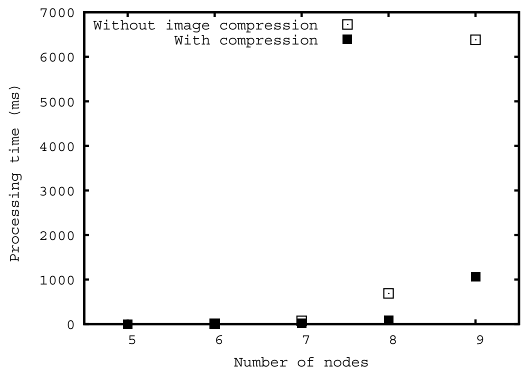

Figure 9. Relations between processing time and the number of nodes.

We have implemented and estimated the procedure to show usefulness of it. As results of estimation, it is verified that the proposed scheme is effective to solve the TSP.

REFERENCES

1. J. Tanida and Y. Ichioka, "A paradigm for digital optical computing based on coded pattern processing," *Intl. J. Opt. Comput.* **1**, pp. 113–128, 1990.
2. T. Haist and W. Osten, "An optical solution for the traveling salesman problem," *Optics Express* **15**, pp. 10473–10482, 2007.
3. N. T. Shaked, S. Messika, S. Dolev, and J. Rosen, "Optical solution for bounded np-complete problem," *Appl. Opt.* **46**, pp. 711–724, 2007.
4. K. Nitta, Y. Tado, O. Matoba, and T. Yoshimura, "Simulation method for quantum algorithm using optical array logic," *in Technical digest in OSA Topical Meeting on CD-ROM* **JWB8**, 2005.
5. K. Nitta, Y. Tado, O. Matoba, and T. Yoshimura, "A method for factorization by means of digital optical computing and image compression," *SPIE.* **6695**, p. 66950B, 2007.