Optical processor for solving the traveling salesman problem (TSP)

Natan T. Shaked[1], Gil Simon[1], Tal Tabib[1], Stephane Mesika[2],
Shlomi Dolev[2] and Joseph Rosen[1]

[1]Department of Electrical and Computer Engineering and [2]Department of Computer Science
Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel.

ABSTRACT

This paper introduces an optical solution to (bounded-length input instances of) an NP-complete problem called the traveling salesman problem using a pure optical system. The solution is based on the multiplication of a binary-matrix, representing all feasible routes, by a weight-vector, representing the weights of the problem. The multiplication of the binary-matrix by the weight-vector can be implemented by any optical vector-matrix multiplier. In this paper, we show that this multiplication can be obtained by an optical correlator. In order to synthesize the binary-matrix, a unique iterative algorithm is presented. This algorithm synthesizes an N-node binary-matrix using rather small number of vector duplications from the ($N-1$)-node binary-matrix. We also show that the algorithm itself can be implemented optically and thus we ensure the entire optical solution to the problem. Simulation and experimental results prove the validity of the optical method.

Keywords: Optical Computing, Optical Data Processing, Optical Correlators, NP-Complete Problems.

1. INTRODUCTION

NP-complete problems[1] are problems which are usually very hard to solve by using deterministic computational systems. In addition, they are equivalent to each other in the sense that if you could find a way to solve one NP-complete problem within certain time complexity, then you could solve any NP-problem within the same time complexity.

The traveling salesman problem (TSP) describes a salesman who travels among N nodes interconnected by a net of roads. The salesman should not visit the same node twice and has to complete his travel at the minimal possible cost. The cost may be the time it takes him to complete the travel, the money he has to pay for the roads, the energy he wastes, etc. Fig. 1 illustrates an example of a 5-node TSP. The cost of traveling from one node to the other is considered as the interconnecting weight between these two nodes. The interconnecting weights are represented in Fig. 1 by the numbers written at the graph edges.

The TSP is an NP-complete problem and thus its solution is very hard to obtain. In recent years, many heuristic algorithms for solving the TSP have been proposed[1]. However, these algorithms sustain several disadvantages. First, some of these algorithms are not optimal in a way that the solution they obtain may not be the best one. Second, their runtime is not always defined in advance, since for every problem there are certain cases for which the computation time is very long due to unsuccessful attempts for optimization[2]. Therefore, in order to ensure a preliminary-defined computation time, we may prefer to exhaustively check all possible solutions. However, because of the vast amount of possible solutions (and therefore the difficult complexity of the calculation and the large amount of required memory), conventional computers may find it hard to carry out this exhaustive search.

In the current paper, we present a new pure-optical method for exhaustively solving the TSP by using an optical multiplication of a binary-matrix, representing all feasible routes, by a weight-vector, representing the weights of the problem. The solution, achieved by this method within a well-defined computation time, is optimal and ensures that no better solution exists.

The proposed method assumes the existence of the binary-matrix of an N-city TSP and once this matrix is given, all TSPs of N or fewer nodes can be solved. We also suggest an efficient algorithm for the synthesis of this binary-matrix, as well as present its optical implementation and demonstrate it by simulations and lab experiments.

Optical Information Systems IV, edited by Bahram Javidi, Demetri Psaltis, H. John Caulfield,
Proc. of SPIE Vol. 6311, 63110G, (2006) · 0277-786X/06/$15 · doi: 10.1117/12.683979

2. METHODOLOGY

As explained earlier, our solution to the TSP is based on a multiplication of a binary-matrix, representing all feasible routes, by a weight-vector, representing the weights of the problem. In subsection 2.1, we present an efficient algorithm for synthesizing the binary-matrix, whereas in subsection 2.2, we explain how to obtain the TSP solution by performing the abovementioned vector-matrix multiplication.

2.1 Synthesis of the binary-matrix

In Ref. 3, we have proposed a unique algorithm for synthesizing the binary-matrix of an N-node TSP by using the binary-matrix of an (N−1)-node TSP. The main advantage of this algorithm is that it uses duplications of large vectors (in the order of the number of the feasible routes) by employing a relatively small amount of repetitions (in the order of the weight-vector length). As we show later on, this algorithm can be implemented by a pure optical system.

The binary-matrix of an N-node TSP contains (N−1)! rows, each of which represents a different route, and N(N−1) columns, each of which represents a different edge connecting one node to another. A '1' in the k-th row and in the l-th column of the binary-matrix means that the k-th route contains the l-th edge.

The iterative algorithm for synthesizing the binary-matrix starts with a binary-matrix representing the case of a 3-node TSP and extends this matrix iteratively to a binary-matrix of the TSP with the required number of nodes. This algorithm is composed of two stages: the initialization stage and the induction stage. In the initialization stage, the weights (and hence the binary-matrix columns) are arranged in a certain order so that the resulting binary-matrix has some degree of symmetry. According to this order, the weights with their second index as 1 (which are underlined in the following equation) replace the orderly weights $w_{k,k}$:

$$\mathbf{w} = \begin{array}{l} [w_{1,2}, w_{1,3}, w_{1,4}, w_{1,5}, \ldots. w_{1,i}, \ldots w_{1,N}, \\ \underline{w_{2,1}}, w_{2,3}, w_{2,4}, w_{2,5}, \ldots. w_{2,i}, \ldots w_{2,N}, \\ w_{3,2}, \underline{w_{3,1}}, w_{3,4}, w_{3,5}, \ldots. w_{3,i}, \ldots w_{2,N}, \\ w_{4,2}, w_{4,3}, \underline{w_{4,1}}, w_{4,5}, \ldots. w_{2,i}, \ldots w_{2,N}, \\ \qquad \qquad \ldots \\ w_{N,2}, w_{N,3}, w_{N,4}, w_{N,5}, \ldots. w_{N,i} \ldots w_{N,N-1}, \underline{w_{N,1}}]^T \end{array} \qquad (1)$$

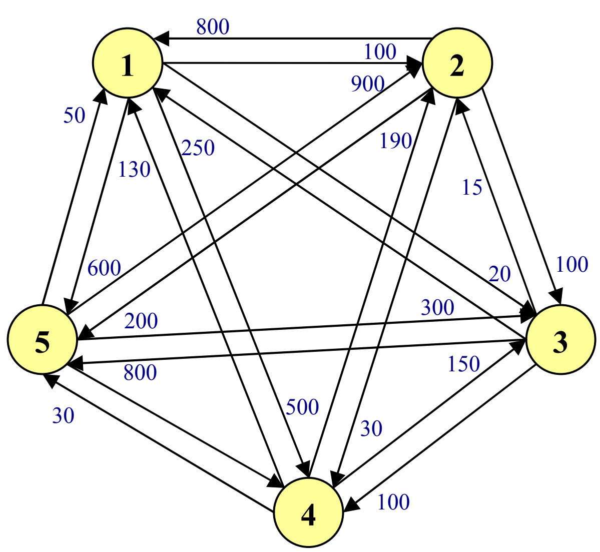


Fig. 1. Example of a 5-node TSP.

Next, a binary-matrix containing the two feasible routes passing through 3 nodes is generated as follows:

$$\mathbf{b}_{N=3} = \begin{matrix} & ref & w_{1,2} & w_{1,3} & w_{2,1} & w_{2,3} & w_{3,2} & w_{3,1} \\ T_1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ T_2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{matrix} \quad , \tag{2}$$

where T_k indicates the binary-matrix row which represents the k-th route and $w_{i,j}$ represents the weight of the path connecting node i and node j. Note that the left column in this matrix, marked by the sign "*ref*", is a reference column (which is utilized later in this subsection) and should not be considered when analyzing the routes represented in the matrix. As can be seen from the binary-matrix in Eq. (2), the first route T_1 is: Node 1 → Node 2 → Node 3 → Node 1, whereas the second route T_2 is: Node 1→ Node 3 → Node 2 → Node 1.

In the induction stage of the transition from an (N−1)-node TSP to an N-node TSP, we start by defining a new binary-matrix in the size of $[(N-1)!]\times[N(N-1)+1]$ and while the first column is reserved for the reference column, the rest of the $N(N-1)$ columns is reserved for the columns which are related to the weights. The new matrix is then divided into $N-1$ horizontal sections, each of which contains $(N-2)!$ rows. Each of the columns (except the reference column) is duplicated once from the source-matrix (the binary-matrix of an (N−1)-node TSP) into each of the sections of the target-matrix (the binary-matrix of an N-node TSP), whereas the reference column is duplicated twice into each of the sections. The duplication of the columns from the source-matrix into the target-matrix, as demonstrated in Fig. 2, is always performed by the same set of rules described below:

1. Duplication of the first (reference) column of the source-matrix:

   a. Duplicate the first (reference) column of the source-matrix into the left column of each of the $N-1$ sections of the target-matrix. This generates the new reference column in the target-matrix. This rule is demonstrated by the dashed arrows in Fig. 2.

   b. Duplicate the first (reference) column of the source-matrix into the columns of the target-matrix related to the weights $w_{1,k+1}$, where k is the section number. This means to duplicate this source-matrix column into the column related to the weight $w_{1,2}$ in the first section of the target-matrix, into the column related to the weight $w_{1,3}$ in the second section of the target-matrix and so on until it is duplicated into the column related to the weight $w_{1,N}$ in the last section of the target-matrix. This rule is demonstrated by the dashed-dotted arrows in Fig. 2.

2. Duplication of each of the remaining $(N-1)(N-2)$ columns of the source-matrix:

   a. Fill the first section of the target-matrix: each time take a different column related to the weight $w_{i,j}$ in the source-matrix and duplicate it into the column related to the weight $w_{m,n}$ in the first section of the target-matrix following these rules: if $j=1$, then $m=i+1$ and $n=1$. This rule is demonstrated by the thick solid arrows in Fig. 2. Otherwise, if $j \neq 1$, then $m=i+1$ and $n=j+1$. This rule is demonstrated by the thick dotted arrows in Fig. 2.

   b. Fill the remaining sections of the target-matrix: each time take a different column related to the weight $w_{i,j}$ in the source-matrix and duplicate it into the column related to the weight $w_{m,n}$ in the k-th section of the target-matrix ($k \geq 2$) following these two-step rules: first, if $j=1$, then $m'=i+1$ and $n'=1$. This rule is demonstrated by the thin solid arrows in Fig. 2. Otherwise, if $j \neq 1$, then $m'=i+1$ and $n'=j+1$. This rule is demonstrated by the thin dotted arrows in Fig. 2. Second, if $m'=2$, then $m=k+1$; if $m'=k+1$, then $m=2$. Otherwise, if $m' \neq 2$ and $m' \neq k+1$, then $m=m'$. The same goes for n' and n: if $n'=2$, then $n=k+1$; if $n'=k+1$, then $n=2$. Otherwise, if $n' \neq 2$ and $n' \neq k+1$, then $n=n'$.

3. Fill the unfilled positions in the target-matrix with zeros.

The above-described rules should be implemented for the transition from the $N=3$ binary-matrix to the $N=4$ binary-matrix, for the transition from the $N=4$ binary-matrix to the $N=5$ binary-matrix and so on until reaching the binary-matrix with the required number of nodes.

Let us compute the complexity of the induction stage, which determines the complexity of the problem. According to phases 1a and 1b of the induction stage of the algorithm, the number of duplications required for the reference column is $2(N-1)$, since we have $N-1$ sections in the target-matrix. The number of columns needed to be duplicated in phases 2a and 2b is (N−2)(N−1) (the number of the rest of the columns in the source-matrix) and they are duplicated into all of the sections, which means N−1 times. Therefore, the number of the required duplications is:

$$\#Dup^{(1)}_{(N-1)\to N} = 2(N-1) + (N-1)^2(N-2) = N^3 - 4N^2 + 7N - 4 \approx \mathrm{O}(N^3) \tag{3}$$

As this process is recursive and since we start with the binary-matrix of a 3-node TSP, the total complexity of the induction stage is:

$$\#Dup^{(1)}_{3\to N} = \sum_{k=4}^{N}\left[2(k-1) + (k-1)^2(k-2)\right] \approx \mathrm{O}(N^4) \tag{4}$$

If we assume that the duplication of each source-matrix columns into the different sections of the target-matrix can be done simultaneously, then according to phases 1a and 1b of the induction stage of the algorithm, the number of duplications required for the reference column is 1 (since it is duplicated into the target-matrix 2(N−1) times simultaneously). The number of columns needed to be duplicated in phases 2a and 2b is (N−2)(N−1) (the number of the rest of the columns in the source-matrix) and they are duplicated into all of the (N−1) sections of the target-matrix simultaneously. Therefore, the number of the required duplications is:

$$\#Dup^{(2)}_{(N-1)\to N} = 1 + (N-1)(N-2) = N^2 - 3N + 3 \approx \mathrm{O}(N^2) \tag{5}$$

And the number of duplications required for the transition from a 3-node TSP into an N-node TSP is given by:

$$\#Dup^{(2)}_{3\to N} = \sum_{k=4}^{N}\left[1 + (k-1)(k-2)\right] \approx \mathrm{O}(N^3) \tag{6}$$

2.2 Vector-matrix multiplication for obtaining the TSP solution

As explained before, once the N-node binary-matrix is synthesized, it can be used to solve TSP with N or fewer nodes. In order to obtain the TSP solution, we multiply the synthesized binary-matrix by a weight-vector, representing the TSP weights. The resulting product is a vector containing the lengths of the TSP routes. This can be expressed mathematically by the following formula:

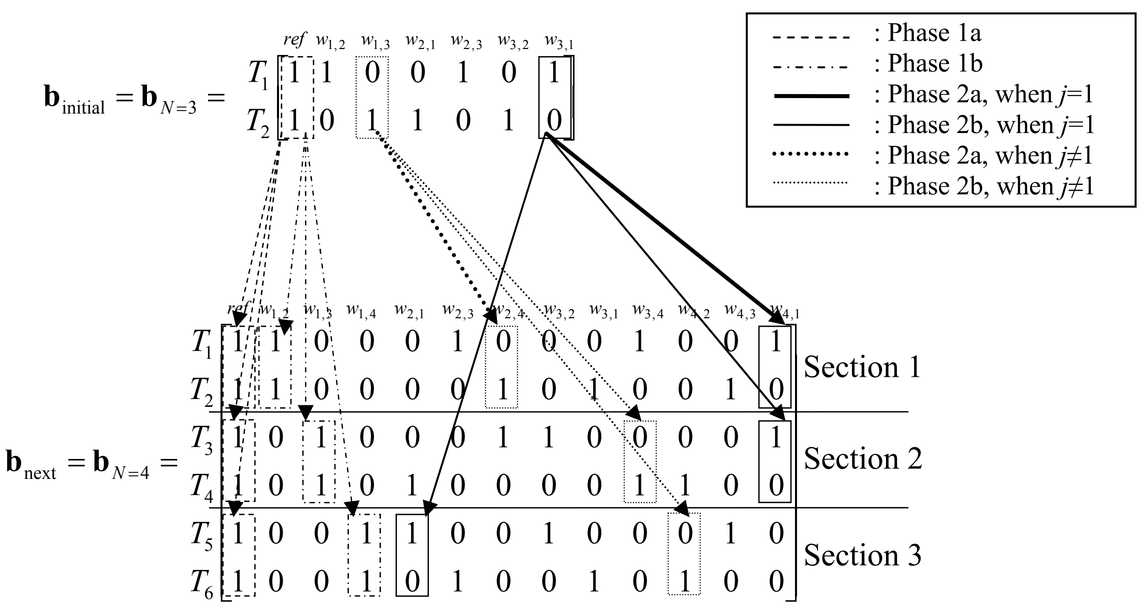


Fig. 2. Example of the transition from the binary-matrix of a 3-node TSP to the binary-matrix of a 4-node TSP according the binary-matrix algorithm.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} w_{1,2} & w_{1,3} & w_{2,1} & w_{2,3} & w_{3,2} & w_{3,1} \end{bmatrix}^T = \begin{bmatrix} w_{1,2} + w_{2,3} + w_{3,1} \\ w_{1,3} + w_{3,2} + w_{2,1} \end{bmatrix} \tag{7}$$

As can be seen from this equation, in this case the resulting length-vector contains two elements. Each element is the total length of the corresponding route. Note that although the above-described example is quite simple, the same method can be carried out for any N-node TSP and the resulting vector contains $(N–1)!$ elements. After obtaining the length-vector, its minimal element coincides with the best route. Note that other related NP-complete problems can be solved using this method. For example, the Hamiltonian path problem (HPP), which is a decision problem of finding out whether there is a route that does not contain the same node more than once and connects all nodes in the graph, can also be solved by the proposed optical method. This can be carried out by multiplying the synthesized binary-matrix by a binary weight-vector (which implies that some of the edges may be blocked). Then, we just have to see if there is an element which is not equal to zero in the resulting product vector and this means that a Hamiltonian path is found.

3. OPTICAL IMPLEMENTATION

In section 2, we describe a method for solving the TSP using a multiplication of a binary-matrix by a weight-vector. In this section, we show how this method can be optically implemented. Our design uses the benefits of optics in order to perform the multiplication of the quite large binary-matrix by the weight-vector. This is, of course, hard to do by conventional computers due to the vast amount of memory storage and the complexity of the calculation. Subsection 3.1 presents the optical implementation of the binary-matrix synthesis according to the binary-matrix algorithm proposed in subsection 2.1, whereas subsection 3.2 presents the optical implementation of the vector-matrix multiplication for obtaining the length-vector of the TSP, as explained in subsection 2.2.

3.1 Optical synthesis of the binary-matrix

The algorithm proposed in subsection 2.1 can be used in order to optically synthesize the binary-matrix. The transition to the binary-matrix of an N-node TSP requires the existence of the binary-matrix of an $(N–1)$-node TSP. This transition is based on duplications of large vectors. The size of the duplicated vectors is $(N–2)!$ and the complexity of the duplication is given by Eqs. (4) or (6). The fact that large vectors can be duplicated by optical means, leads us to choose optics for implementing the binary-matrix algorithm.

The $4f$ optical system[4] shown in Fig. 3 is capable of performing a convolution between two images. This system is utilized to carry out a vector duplication by performing a convolution between the vector and shifted spatial delta (point) functions. The result of this convolution is the input column shifted to the locations of the delta functions. We use pairs of symmetric delta functions in order to get a real spectral transform that can be represented on a regular slide or on a spatial light modulator (SLM).

The binary-matrix is represented on a slide (or an SLM). A white rectangle on the slide represents a '1' in the binary-matrix and a black rectangle on the slide represents a '0' in this matrix. Fig. 3 demonstrates the transition from the 4-node TSP binary-matrix to the 5-node TSP binary-matrix. As shown in Fig. 3, the source-matrix (the binary-matrix of $N–1 = 4$ nodes) is represented on the slide placed in plane P_1, whereas the target-matrix (the binary-matrix of $N = 5$ nodes) is accumulated during the correlation iterations on a film (or CCD camera) which is placed on plane P_3. As shown in Fig. 3, the column that we would like to duplicate (according to the binary-matrix algorithm) is extracted from the source-matrix by a vertical slit.

The synthesized transformed mask of the shifted delta functions is placed on a slide (or an SLM) in plane P_2. Each delta function is shifted in both the vertical and horizontal directions. The vertical shift is proportional to the suitable target-matrix section (according to the binary-matrix algorithm). The horizontal shift is composed of both the shift of the column in the desired horizontal direction according to the binary-matrix algorithm and an additional shift which assures that the target-matrix will not overlap with the other spatial components appearing on plane P_3. This undesired overlap may occur because the convolution with the delta functions yields two duplications: one in the positive direction from the center of plane P_3, and the other in the negative direction. At the end of the process, plane P_3 contains the binary-matrix of an N-node TSP. Afterwards, we can utilize this matrix to synthesize the binary-matrix of an $(N+1)$-node TSP by using the same method. This continues till reaching the binary-matrix of the TSP with the desired number of nodes.

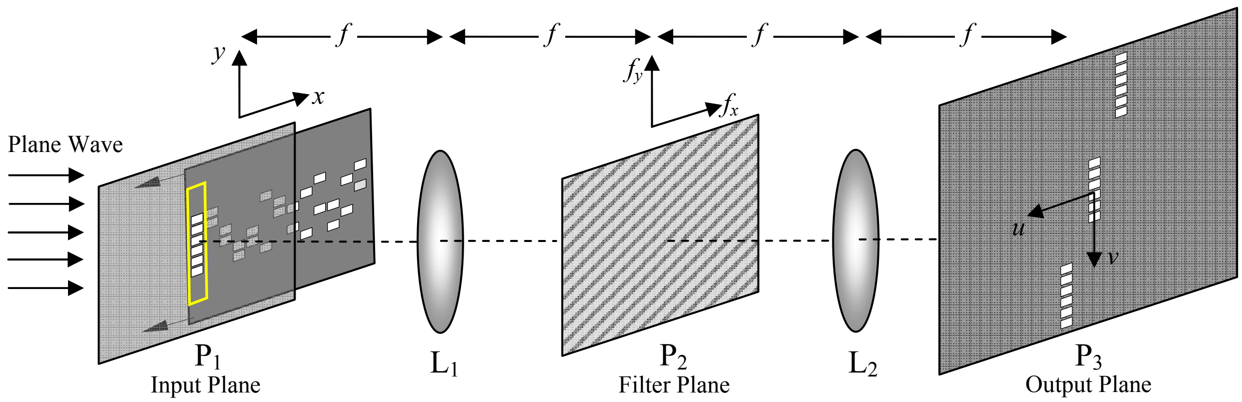


Fig. 3. 4*f* optical system for the optical implementation of the binary-matrix algorithm.

The algorithm described in subsection 2.1 provides the destination of the duplications for each of the source-matrix columns. This can be accomplished by both the first and second methods, the complexities of which are given by Eqs. (4) and (6), respectively. In the first method, we perform a duplication of each of the source-matrix columns into a single destination in the target-matrix. In order to optically perform that, we convolve two shifted symmetric delta functions with the column which should be duplicated. The transformed representation of these two symmetric delta functions can be expressed on the spectral domain as follows:

$$H^{(1)} = 1 + \cos\left(2\pi f_x (X + A) + 2\pi f_y Y\right), \tag{8}$$

where f_x and f_y are the horizontal and vertical spatial frequencies respectively, X and Y are the horizontal and vertical shifts respectively, given by the binary-matrix algorithm, and A is the horizontal shift which assures the separation of the two matrices that appear on plane P_3.

In the second method, we perform a duplication of each of the source-matrix columns into multiple destinations in the target-matrix simultaneously. In order to optically perform that, we convolve a set of shifted symmetric delta function pairs with the column which should be duplicated. The transformed representation of this set of shifted symmetric delta function pairs can be given by the Burch method[5] as:

$$H^{(2)} = bias + \Im\left\{\sum_i \left[\delta\left(x - (X_i + A), y - Y_i\right) + \delta\left(x + (X_i + A), y + Y_i\right)\right]\right\}. \tag{9}$$

where X_i and Y_i are the horizontal and vertical shifts, respectively, given by the binary-matrix algorithm for the *i*-th delta function pair.

3.2 Optical vector-matrix multiplication for obtaining the TSP solution

In order to perform the vector-matrix multiplication, the same 4*f* optical system used in subsection 3.1 can be put into action again. This time, as shown in Fig. 4, the weight-vector is represented on a slide (or an SLM) placed in the input plane P_1 by a set of normalized gray-scale rectangles[3], whereas the transformed binary-matrix mask is represented on a slide (or an SLM) placed in the filter plane P_2. In order to synthesize the transformed binary-matrix mask, we can use several methods, such as the Burch method[5], VanderLugt method[4,6], etc. The output plane P_3 is the correlation plane of the system and it contains a correlation matrix in which the middle column represents the desired multiplication of the binary-matrix by the weight-vector. This multiplication product is the length-vector of the TSP. In Ref. 3, we use the joint transform correlator (JTC)[4,7] in order to carry out the multiplication. This is performed in Ref. 3 by using both simulations and lab experiments. In the current paper, we use the 4*f* optical system in order to carry out the multiplication. The binary-matrix is transformed to the spectral domain by using the Burch method. This is demonstrated in the current paper by simulations. Simulations and experimental results demonstrating the optical synthesis of the binary-matrix, which is explained in subsection 3.1, are also given in the current paper.

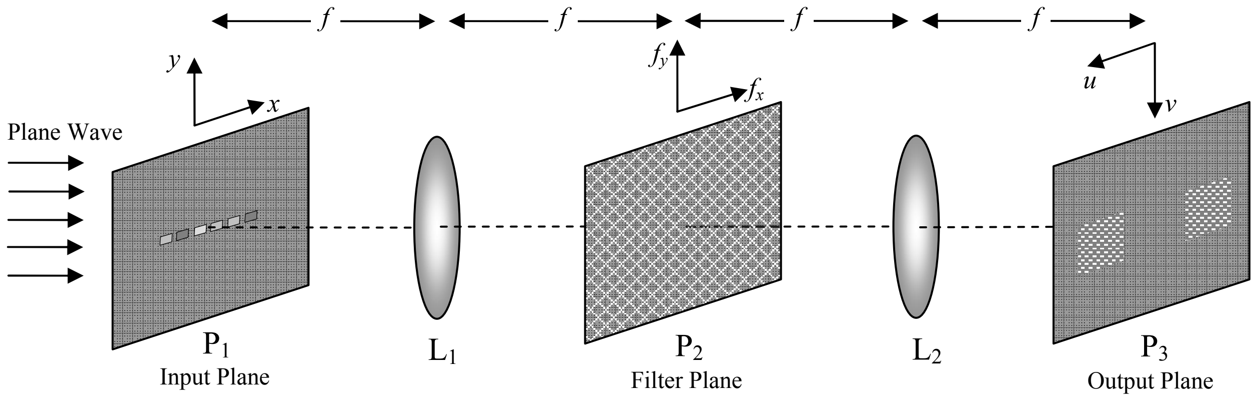


Fig. 4. 4*f* optical system for performing the multiplication of the binary-matrix by the weight-vector.

4. SIMULATION RESULTS

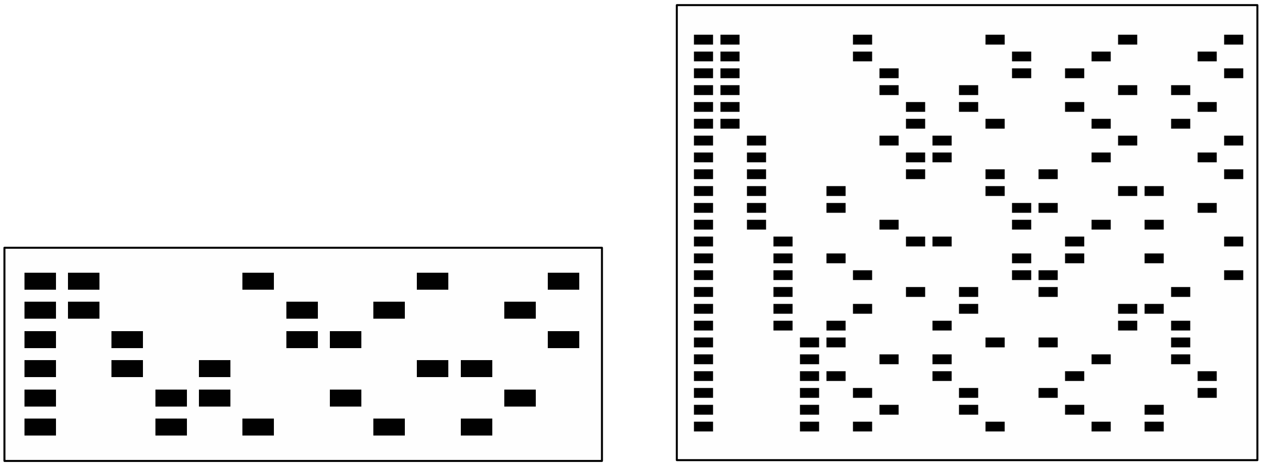
This section presents the simulations performed in order to check the proposed optical method. In the first simulation (presented in subsection 4.1), we demonstrate the synthesis of the binary-matrix for the transition from the binary-matrix of a 4-node TSP to the binary-matrix of a 5-node TSP, whereas in the second simulation (presented in subsection 4.2), we demonstrate how the synthesized binary-matrix can be used to solve the 5-node TSP shown in Fig. 1.

4.1 Simulation of the optical synthesis of the binary-matrix

In order to synthesize the binary-matrix of a 5-node TSP, we assume the existence of a binary-matrix of a 4-node TSP (which can be synthesized beforehand from the binary-matrix of a 3-node TSP by using the same set of rules, defined in subsection 2.1). To do that, we simulate the 4*f* optical system illustrated in Fig. 3. On the input plane P_1, we place the binary-matrix of a 4-node TSP (the source-matrix) shown in Fig. 5(a), whereas the accumulated output plane P_3 will eventually contain the binary-matrix of a 5-node TSP (the target-matrix) shown in Fig. 5(b). As demonstrated in Fig. 3, in each iteration only a single column from the source-matrix enters the system. This column is Fourier transformed and then multiplied by the required filter which determines the locations into which this column is duplicated on the output plane P_3. On the filter plane P_2, we place the synthesized filter mask of the transformed delta functions.

Simulations of both methods discussed in subsection 3.1 are presented. In the first simulation, we use the first method in which each of the columns is duplicated into a single location each time. According to Eq. (3), the number of iterations needed for the transition from the abovementioned source-matrix to the abovementioned target-matrix is 56. Thus, 56 different filters (each of which is defined by Eq. (8)) are required in order to perform this transition. Fig. 6 shows the first 8 out of the 56 filters required for the transition. Figs. 7(a), (b), (c) and (d) show the accumulative output plane P_3 after completing the first, second, third and the last (56th) iteration of the transition respectively. In Fig. 7(a), the reference column from the source-matrix is duplicated into the first section of the target-matrix that appears in the right diffraction order of the output plane P_3. Similarly, in Figs. 7(b) and 7(c), the reference column from the source-matrix is duplicated into the second and third sections of the target-matrix respectively. Eventually, as shown in Fig. 7(d), the right diffraction order of the output plane P_3 contains the complete target-matrix.

In the second simulation, we use the second method in which each of the columns is duplicated into multiple locations each time. According to Eq. (5), the number of iterations needed for this transition is 13. Thus, only 13 different filters (each of which is defined by Eq. (9)) are required in order to perform the transition. Fig. 8 shows these filters. Figs. 9(a), (b), (c) and (d) show the accumulative output plane P_3 after completing the first, second, third and the last (13th) iteration of the transition respectively. In Fig. 9(a), the reference column from the source-matrix is duplicated into the first, second, third and fourth sections of the target-matrix simultaneously (twice into each section). In Figs. 9(b) and 9(c), the second and third columns, respectively, are simultaneously duplicated from the source-matrix into each of the sections of the target matrix (once into each section). Eventually, as shown in Fig. 9(d), the right diffraction order of the output plane P_3 contains the final target-matrix (the same result achieved in the first method, as shown in Fig. 7(d)).



(a) (b)

Fig. 5. Simulation results (contrast-inverted pictures): (a) the binary-matrix of a 4-node TSP (the source-matrix); (b) the binary-matrix of a 5-node TSP (the target-matrix).

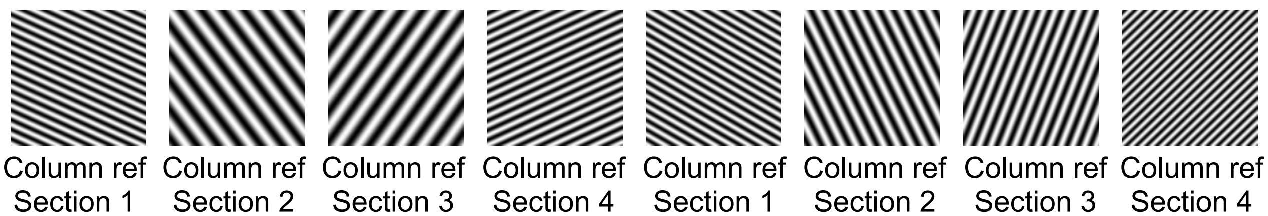
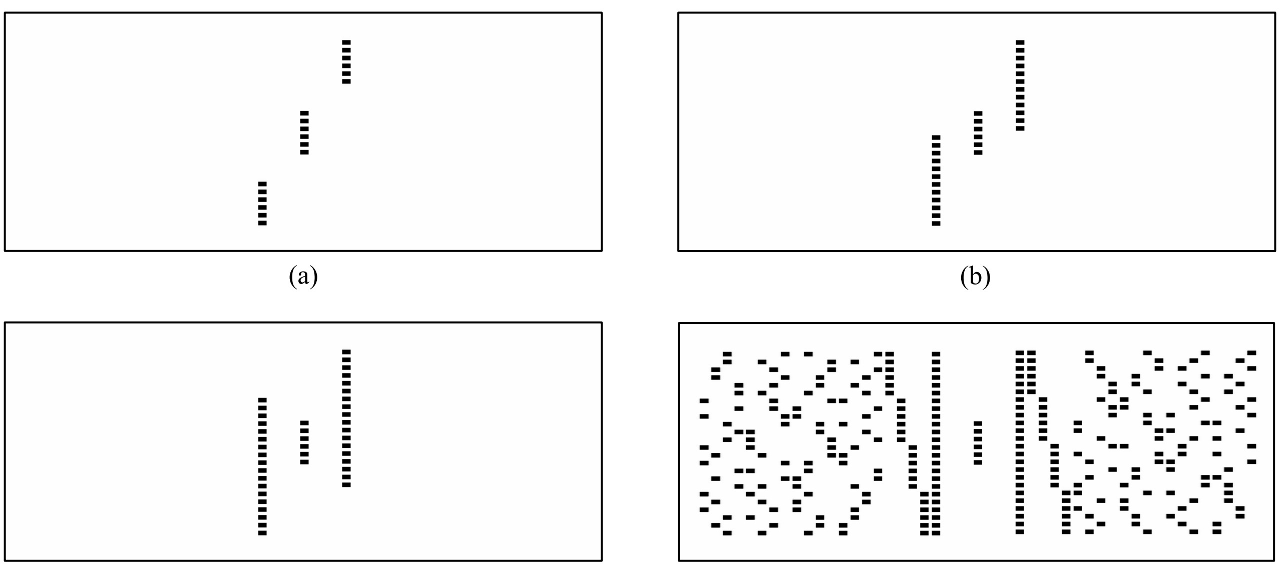


Fig. 6. The first 8 out of 56 filters used for the transition from the binary-matrix of a 4-node TSP to the binary-matrix of a 5-node TSP in the first method (duplicating into a single location each time). The source-matrix column tag and the target-matrix section number are written below each filter.



(c) (d)

Fig. 7. Four out of 56 contrast-inverted accumulative correlation planes depicting the transition from the binary-matrix of a 4-node TSP to the binary-matrix of a 5-node TSP in the first method (duplicating into a single location each time): (a) First duplication; (b) Second duplication; (c) Third duplication; (d) Last (56th) duplication.

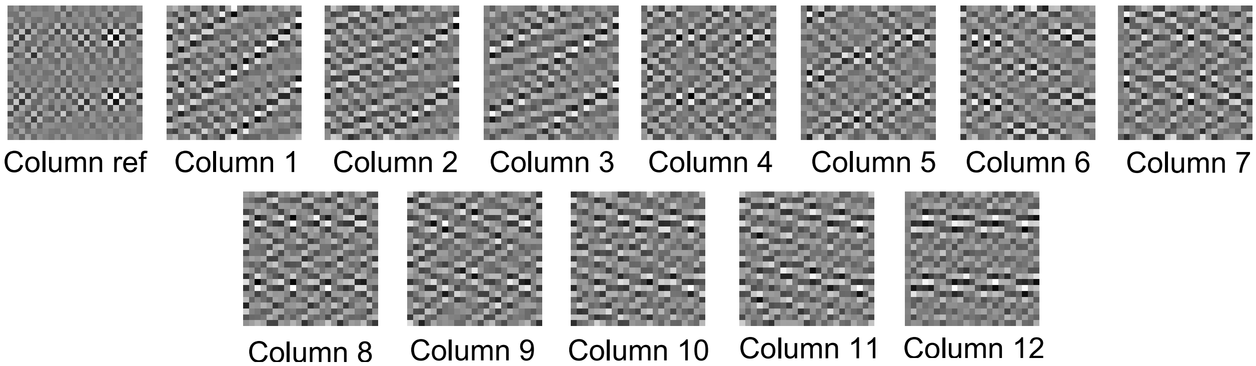


Fig. 8. The 13 filters used for the transition from the binary-matrix of a 4-node TSP to the binary-matrix of a 5-node TSP in the second method (duplicating into multiple locations each time). The source-matrix column tag is written below each filter.

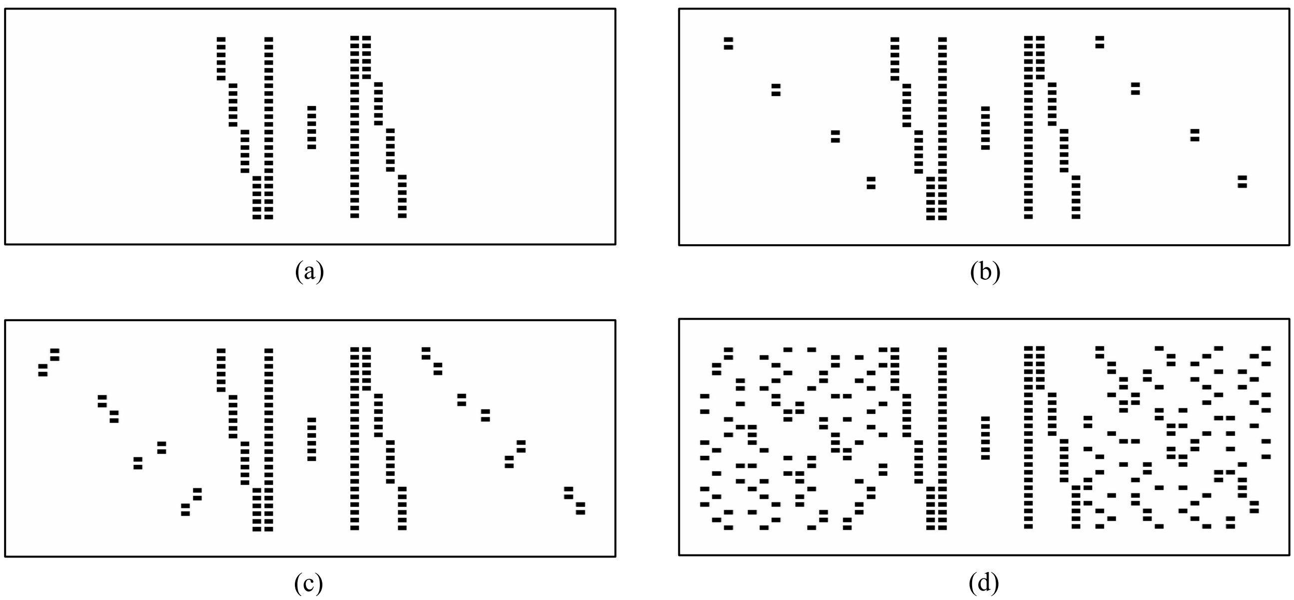


Fig. 9. Four out of 13 contrast-inverted accumulative correlation planes depicting the transition from the binary-matrix of a 4-node TSP to the binary-matrix of a 5-node TSP in the second method (duplicating into multiple locations each time): (a) First duplication; (b) Second duplication; (c) Third duplication; (d) Last (13th) duplication.

For both methods, the required binary-matrix of a 5-node TSP (the target-matrix) can be easily cut out from the right diffraction order of the output plane P_3 that appears in Figs. 7(d) or 9(d). The left diffraction order matrix is an abnormal binary-matrix caused by the inverse duplication of the delta function, due to the Burch method and owing to the fact that not all of the columns are symmetric.

Once the binary-matrix of a 5-node TSP is obtained, we can use it to either solve any TSP of 5 or fewer nodes, or use it to synthesize the binary-matrix of a 6-node TSP by utilizing the same set of rules (defined in subsection 2.1)

4.2 Simulation of the optical vector-matrix multiplication for obtaining the TSP solution

In this subsection, we demonstrate the solution of the TSP shown in Fig. 1. This is performed (according to the technique explained in subsection 3.2) by correlating the suitable gray-scale weight-vector with the binary-matrix of a 5-node TSP (Fig. 5(b)) that is synthesized in the former subsection. This gray-scale weight-vector, placed on the input plane P_1 in the

4f optical system illustrated in Fig. 4, is shown in Fig. 10(a). The Burch mask of the synthesized binary-matrix, used as the filter (plane P_2) of the 4f optical system illustrated in Fig. 4, is shown in Fig. 10(b). The result of the correlation operation appears on the output plane P_3 of the 4f optical system illustrated in Fig. 4. This output plane is shown in Fig. 10(c) and it contains two correlation matrices, the right matrix of which can be used for the determination of the best route. This route is indicated by the darkest spot in the middle column of this correlation matrix. The peak heights

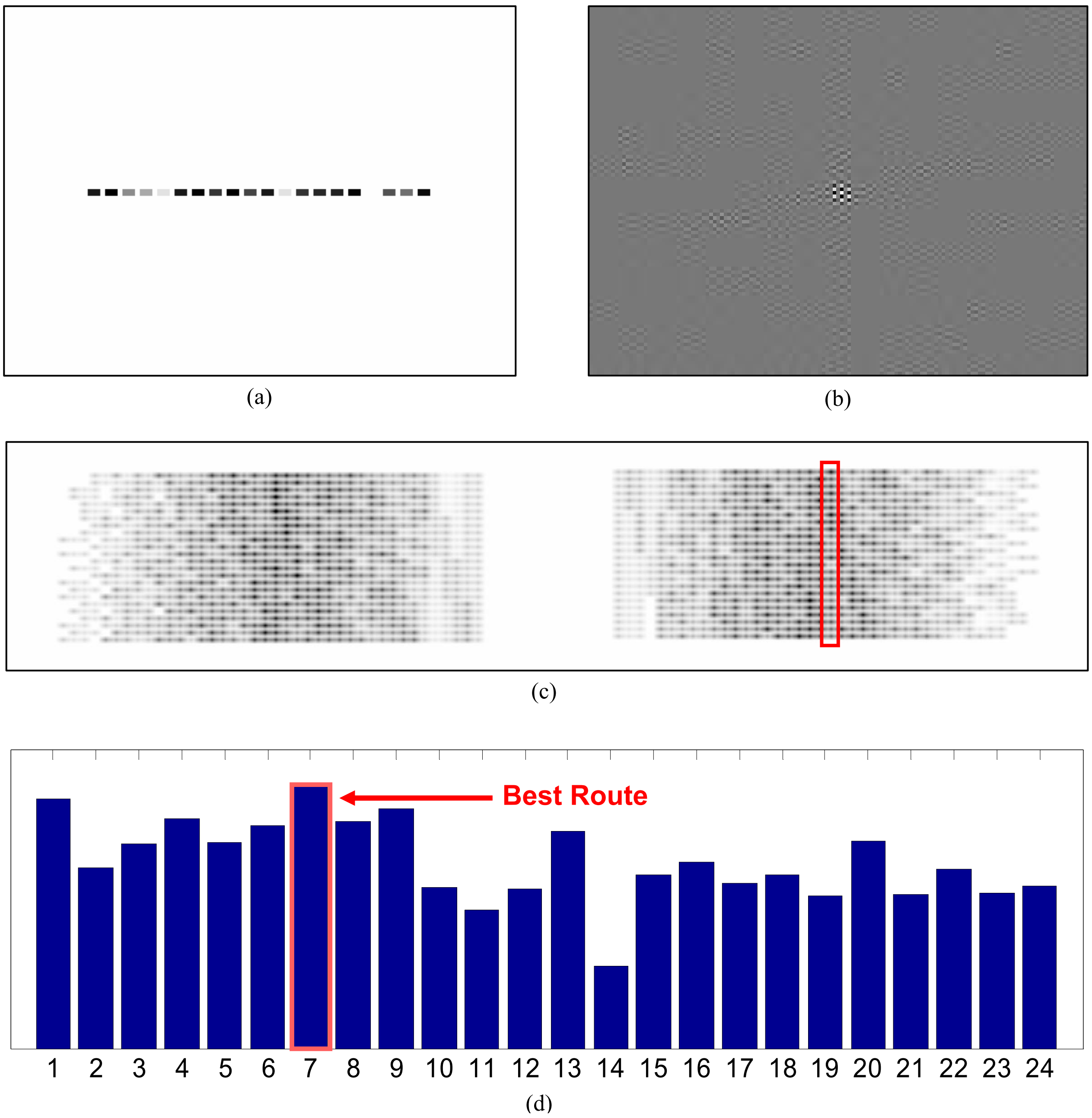


Fig. 10. Simulation results of multiplying the binary-matrix by the weight-vector in order to obtain the TSP solution: (a) The contrast-inverted weight-vector corresponding to the 5-node TSP in Fig. 1; (b) The Burch mask of the synthesized binary-matrix in Fig. 5(b); (c) The contrast-inverted correlation plane containing two diffraction orders, the right order of which is the correlation matrix; (d) Bars representing the peaks across the middle column of the correlation matrix.

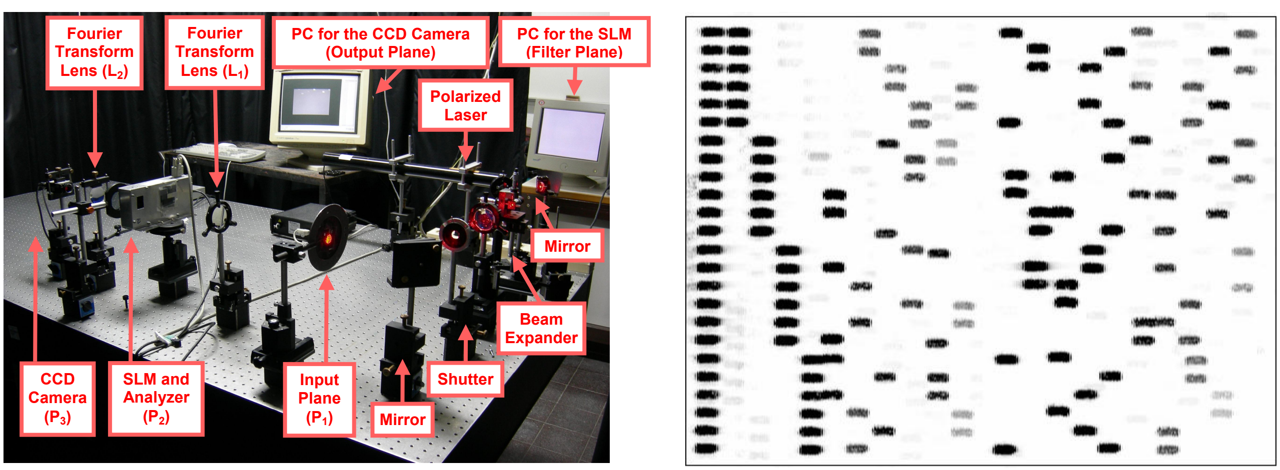
across this middle column are displayed by bars in Fig. 10(d). As seen in this figure, the highest bar appears in the place representing the 7th route. This is the shortest route and thus the solution to the TSP. Going back to the binary-matrix (shown in Fig 5(b)) reveals that this route contains the following weights: $w_{1,3}, w_{2,4}, w_{3,2}, w_{4,5}, w_{5,1}$, which means that the shortest TSP route is: Node 1 → Node 3 → Node 2 → Node 4 → Node 5 → Node 1. As can be seen from Fig. 1, this is indeed the shortest route.

5. EXPERIMENTAL RESULTS

In this section, we demonstrate by an experiment the synthesis of the binary-matrix of a 5-node TSP (the target-matrix) by using the binary-matrix of a 4-node TSP (the source-matrix). This is performed by implementing the first method (demonstrated by a simulation in subsection 4.1), in which each column from the source matrix is duplicated into a single location in the target-matrix each time. Fig. 11(a) shows a photograph of the experiment setup. As can be seen in this figure, a laser (Uniphase 1144/P, 17mW, 632.8nm, HeNe polarized laser) beam is expanded using a beam expander and illuminates the input plane (P_1 in Fig. 3), which is accomplished in the experiment by a regular slide. A lens with a focal length of 25cm Fourier transforms the source-matrix column and the Fourier transform is multiplied by the filter plane (P_2 in Fig. 3), which is accomplished in the experiment by a computer-controlled SLM (CRL Opto XGA2, 1024×768 pixels). 56 filters are projected on the SLM (the first several ones of which are shown in Fig. 6). Then, another lens with a focal length of 30cm Fourier transforms the multiplication result, and the output plane (P_3 in Fig. 3) contains the duplication of the source-matrix column into the suitable location in the target-matrix. A CCD camera (Sony XC75-CE) is placed in the output plane and records the intensity distribution there. The accumulative output plane, composed of the summation of the 56 resulting correlation planes, is shown in Fig. 11(b). As seen in this figure, this plane indeed contains the target-matrix (the binary-matrix of a 5-node TSP), and it can be compared to the accumulated correlation plane shown in Fig. 5(b), obtained by simulation. Note that since the CCD camera aperture is not large enough to contain the binary-matrix of a 5-node TSP, we performed the task by concatenating two CCD camera planes.

6. CONCLUSION

We have proposed an optical method for solving (bounded-length input instances of) NP-complete problems, such as the TSP and the HPP. The method exhaustively checks all feasible solutions. There is a need to solve this kind of problems by an exhaustive search in order to ensure a preliminary-defined solution time. According to the proposed method, we multiply a binary-matrix, representing all feasible routes, by a weight-vector, representing the weights of the problem.



(a) (b)

Fig. 11. (a) The optical experiment setup. (b) The accumulated binary-matrix of a 5-node TSP obtained experimentally.

We have also provided an efficient algorithm for the synthesis of the binary-matrix. Once this matrix is synthesized, it can be used to solve all TSPs and HPPs with the same number or fewer nodes. The synthesis of the binary-matrix is demonstrated by both computer simulations and an optical experiment. There is a full agreement between the simulation and the experimental results. Currently, it is feasible to exhaustively solve TSPs and HPPs which contains 15 or fewer nodes by a single iteration of the proposed optical method within nanoseconds (can be considered as real-time performance), whereas a conventional electronic computer can perform this exhaustive search within tens of seconds (cannot be considered as real-time performance). There is still a problem to solve, within a single optical iteration, TSPs and HPPs with more than 15 nodes due to the large size of their binary-matrices. Decreasing the wavelength might help reduce the size of the binary-matrix and thus enable the solutions of larger TSPs and HPPs. The decrease of the wavelength is currently quite limited due to the currently available light sources and SLMs. Anyway, in our opinion the real-time performances of the system, which can be obtained for small TSPs and HPPs (till 15 nodes) by using the currently available technologies, signify the advantages of the optical system. A possible direction of future research is to extend the proposed method to solving other NP-complete problems or other difficult problems.

Acknowledgments: This research was supported by the Israel Science Foundation Grant 119_03 as well as by Intel, Deutsche Telecom, Rita Altura Trust Chair in Computer Sciences and the Israel Ministry of Science.

REFERENCES

1. M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, New York, 1979.
2. D. Gutfreund, R. Shaltiel, and A. Ta-Shma, "If NP languages are hard on the worst-case then it is easy to find their hard instances," in *Proc. of the 20th Annual Conference on Computational Complexity* (CCC), San Jose, California 2005. A long version of this paper is available on web in: www.cs.tau.ac.il/~amnon/Papers/GST.cc05.journal.ps
3. N. T. Shaked, S. Messika, S. Dolev, and J. Rosen, "Optical solution for bounded NP-complete problems," Submitted for Publication in *Appl. Opt.*, February 2006.
4. J. W. Goodman**,** *Introduction to Fourier Optics*, 2nd edition, pp. 232-243, McGraw-Hill, New York, 1996.
5. J. J. Burch, "Computer algorithm for the synthesis of spatial frequency filters," *Proc. IEEE* **55**, 599-601 (1967).
6. A. VanderLugt, "Signal detection by complex spatial filtering," *IEEE Trans. on Inf. Theory* **IT-10**, 139-145 (1964).
7. C. S. Weaver and J. W. Goodman, "A technique for optically convolving two functions," *Appl. Opt.* **5**, 1248-1249 (1966).