Optical binary-matrix synthesis for solving bounded NP-complete combinatorial problems

Natan T. Shaked, MEMBER SPIE
Ben-Gurion University of the Negev
Department of Electrical and Computer Engineering
P.O. Box 653
Beer-Sheva 84105, Israel
E-mail: natis@ee.bgu.ac.il

Tal Tabib
Gil Simon
Ben-Gurion University of the Negev
Department of Electrical and Computer Engineering
P.O. Box 653
Beer-Sheva 84105, Israel

Stephane Messika
Shlomi Dolev
Ben-Gurion University of the Negev
Department of Computer Science
P.O. Box 653
Beer-Sheva 84105, Israel

Joseph Rosen, MEMBER SPIE
Ben-Gurion University of the Negev
Department of Electrical and Computer Engineering
P.O. Box 653
Beer-Sheva 84105, Israel

Abstract. An optical method for synthesizing a binary matrix representing all feasible solutions of a bounded (input length restricted) NP-complete combinatorial problem is presented. After the preparation of this matrix, an optical matrix-vector multiplier can be used in order to multiply the synthesized binary matrix and a grayscale vector representing the weights of the problem. This yields the required solution vector. The synthesis of the binary matrix is based on an efficient algorithm that utilizes a small number of long-vector duplications. These duplications are performed optically by using an optical correlator. Simulations and experimental results are given. © *2007 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.2799086]

Subject terms: optical computing; matrix multiplication; correlators; Fourier optics.

Paper 070027R received Jan. 15, 2007; revised manuscript received Apr. 24, 2007; accepted for publication May 4, 2007; published online Oct. 24, 2007. This paper is a revision of a paper presented at the SPIE conference an Optical Information Systems IV, Aug. 2006, San Diego, Calif. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 6311.

1 Introduction

Efficient solutions to NP-complete combinatorial problems have been the goal of many researches.[1] Nevertheless, none of the researches has succeeded in finding an efficient polynomial-time solution to these problems. Due to the difficulty of solving these problems, many approximation and heuristic methods have been proposed in the literature.[1,2] However, the approximation methods do not always find the best solution within a reasonable computation time. On the other hand, the heuristic methods are able to solve these problems quickly for certain cases only. In fact, the computation time of the heuristic methods may be unexpected and even longer than that of an exhaustive search (checking all possibilities in an exhaustive manner), due to unsuccessful attempts at optimization.[2] Therefore, when there is a need to ensure a predefined computation time, we may prefer to exhaustively check all possible solutions. However, because of the vast number of possible solutions (and therefore the intricacy of the calculation and the large amount of required memory), conventional computers may find it hard to carry out this exhaustive search.

Recently,[3] we have proposed a new optical system design that is capable of solving bounded instances of NP-complete problems, such as the traveling salesman problem (TSP) and the Hamiltonian path problem (HPP), by checking all feasible solutions more efficiently than conventional computers. This design is based on a fast optical matrix-

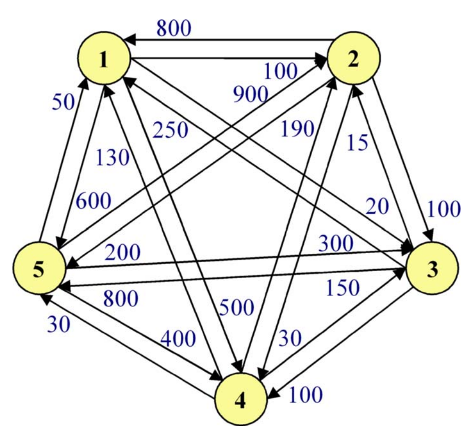


Fig. 1 Example of a five-node TSP.

0091-3286/2007/$25.00 © 2007 SPIE

vector multiplication between a binary matrix, representing all feasible solutions, and a weight vector, representing the problem weights. The multiplication product is a vector representing the solutions of the problem. In the TSP, the required solution is the best (shortest) tour connecting a certain set of given node coordinates, where each node coordinate is visited exactly once. An example of a fully connected TSP graph, containing five nodes, is shown in Fig. 1. In this kind of problem, the matrix-vector multiplication is performed between a binary matrix, representing all feasible TSP tours, and a grayscale weight vector, representing the finite weights between the TSP nodes. The multiplication product is a length vector representing the TSP tour lengths by peaks of light with different intensities. The shortest tour can be found by using an optical polynomial-time binary search, which utilizes an optical threshold plate. On the other hand, in the HPP, a decision whether there is a path connecting two nodes on the HPP graph is required (which implies that some of the graph edges may be blocked). In the HPP, the binary matrix still represents all feasible paths (tours), but the elements of the weight vector in this problem are binary as well. After the matrix-vector multiplication, any peak of light (with a certain intensity) obtained in the output of the optical system means that a Hamiltonian path exists.

The advantage of the proposed method is that once the binary matrix is synthesized, the TSP, HPP, and other related problems of the same order (with the same number of nodes) can be solved optically by only changing the weight vector and performing the matrix-vector multiplication in an optical way. This optical matrix-vector multiplication can be performed by several methods, such as the Stanford multiplier and various correlation methods.[4,5]

In Ref. 3, we have also provided a new efficient algorithm for synthesizing the binary matrix so that it contains all the feasible tours and only them. One advantage of this algorithm is that it synthesizes a binary matrix of N nodes that also contains the binary matrices of fewer than N nodes. This means that this matrix has to be synthesized only once for all problems with N or fewer nodes. Another advantage of this algorithm is that it uses a relatively small number of iterations in order to produce big vectors by duplications of existing vectors.

The synthesized binary matrix contains a large number, $(N-1)!$, of rows, each row representing a different feasible tour, and a relatively small number, $N(N-1)$, of columns, each column representing an edge related to one of the problem weights. Taking into account the large number of rows and the fact that conventional electronic computers may find it hard to duplicate such huge vectors (if the number of nodes, N, is large), the optical system proposed in this paper may be a useful method for synthesizing the binary matrix. According to this optical system design, the binary matrix's long columns are duplicated one after the other by performing a correlation operation with shifted point functions for which the shifts are given by the binary matrix algorithm. We show that the number of duplications needed in order to synthesize a binary matrix of any size can be less than N^3. The proposed method is tested by both simulations and lab experiments.

The rest of the paper is organized as follows. Section 2 introduces the methodology of the proposed method. Section 3 explains the optical implementation of the method. Sections 4 and 5 present simulation and experimental results, respectively. Section 6 makes some concluding remarks.

2 Methodology

For simplicity of explanation, let us refer to the more general case of the TSP (although, as explained earlier, similar problems, such as the HPP, can be solved by the same method). Our solution to the TSP is based on a multiplication of a binary matrix, representing all feasible tours, by a weight vector, representing the weights of the problem. In Sec. 2.1 we present an efficient algorithm for synthesizing the binary matrix, and in Sec. 2.2 we explain how to obtain the TSP solution by performing a matrix-vector multiplication.

2.1 *Synthesis of the Binary Matrix*

The binary-matrix algorithm is able to synthesize 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 numbers on the order of the number of the feasible tours) by employing a relatively small number of repetitions (on the order of the weight-vector length). As we show later on, this algorithm can be implemented in a pure optical system.

The binary matrix of an N-node TSP contains $(N-1)!$ rows, each of which represents a different tour, 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 tour contains the l'th edge.

The iterative algorithm for synthesizing the binary matrix starts with a binary matrix representing the case of a three-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's 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 $w_{k,k}$:

$$
\begin{aligned}
\mathbf{w} = [&w_{1,2}, w_{1,3}, w_{1,4}, w_{1,5}, \ldots, w_{1,i}, \ldots, w_{1,N},\\
&\underline{w_{2,1}}, w_{2,3}, w_{2,4}, w_{2,5}, \ldots, w_{2,i}, \ldots, w_{2,N},\\
&w_{3,2}, \underline{w_{3,1}}, w_{3,4}, w_{3,5}, \ldots, w_{3,i}, \ldots, w_{3,N},\\
&w_{4,2}, w_{4,3}, \underline{w_{4,1}}, w_{4,5}, \ldots, w_{4,i}, \ldots, w_{4,N}, \ldots,\\
&w_{N,2}, w_{N,3}, w_{N,4}, w_{N,5}, \ldots, w_{N,i} \ldots, w_{N,N-1}, \underline{w_{N,1}}]^T.
\end{aligned} \tag{1}
$$

Next, a binary matrix containing the two feasible tours passing through three nodes is generated as follows:

$$
\mathbf{b}_{N=3} = \begin{array}{c} \\ T_1 \\ T_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} \text{ref} & w_{1,2} & w_{1,3} & w_{2,1} & w_{2,3} & w_{3,2} & w_{3,1} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}, \tag{2}
$$

where T_k indicates the binary matrix row that represents the k'th tour, and $w_{i,j}$ represents the weight of the edge connecting node i and node j. Note that the left column in this

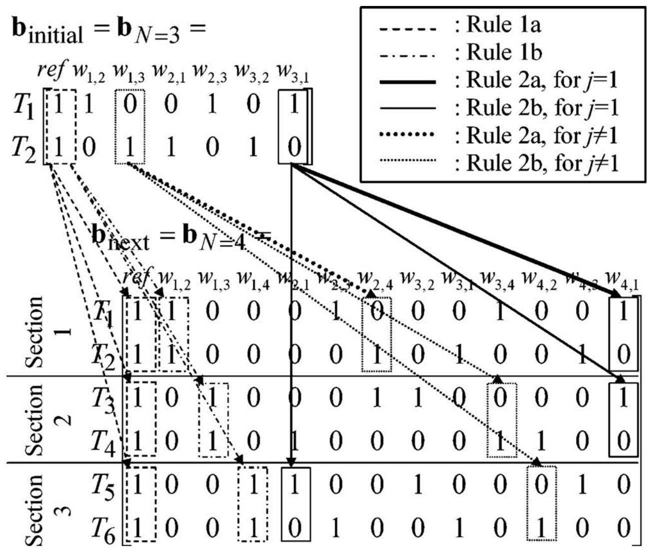


Fig. 2 Example of the transition from the binary matrix of a three-node TSP to the binary matrix of a four-node TSP according to the binary-matrix algorithm.

matrix, marked "ref," is a reference column (which is utilized later in this subsection) and should not be considered when analyzing the tours represented in the matrix. As can be seen from the binary matrix in Eq. (2), the first tour T_1 is node $1 \rightarrow$ node $2 \rightarrow$ node $3 \rightarrow$ node 1, whereas the second tour T_2 is node $1 \rightarrow$ node $3 \rightarrow$ node $2 \rightarrow$ 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 of size $[(N-1)!] \times [N(N-1)+1]$, and while the first column is reserved for the reference column, the rest of the $N(N-1)$ columns are reserved for the columns that 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. Duplicate 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 dash-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 arrow in Fig. 2). Otherwise, if $j \neq 1$, then $m=i+1$ and $n=j+1$. (This rule is demonstrated by the thick dotted arrow 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 ($k \geqslant 2$) 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 preceding 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 rules 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 rules 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 single duplications required for the transition from a binary matrix of an $(N-1)$-node TSP to the binary matrix of an N-node TSP is

$$\begin{aligned}\#\mathrm{Dup}^{(1)}_{(N-1)\rightarrow N} &= 2(N-1) + (N-1)^2(N-2) \\ &= N^3 - 4N^2 + 7N - 4 \approx O(N^3). \qquad (3)\end{aligned}$$

Since this process is recursive, and since we start with the binary matrix of a three-node TSP, the total number of

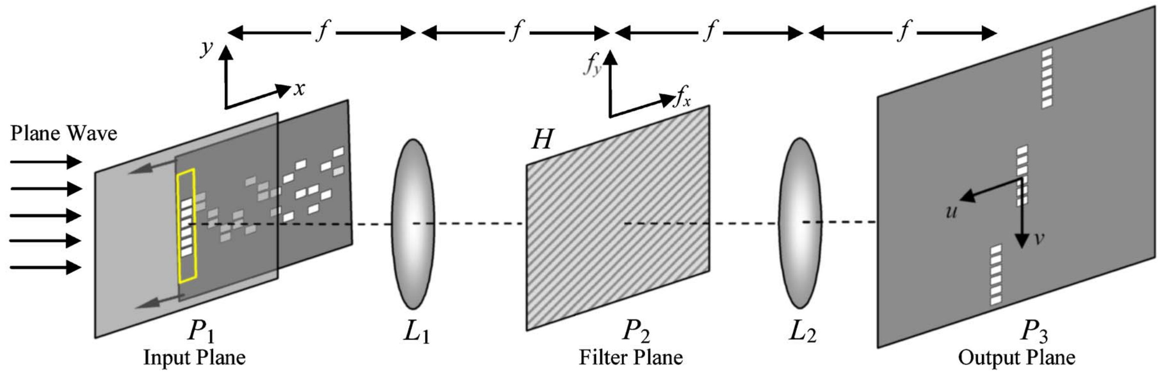


Fig. 3 4f optical system for the optical implementation of the binary-matrix algorithm.

single duplications required for the induction stage (which defines its complexity) is

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

If we assume that the duplication of each of the source-matrix columns into the different sections of the target matrix can be done simultaneously, then according to rules 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 rules 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 multiple duplications required for the transition from a binary matrix of an $(N-1)$-node TSP to the binary matrix of an N-node TSP is

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

and the number of multiple duplications required for the transition from a three-node TSP into an N-node TSP (which defines the complexity of the induction stage) is given by

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

2.2 *Matrix-Vector Multiplication for Obtaining the Problem Solution*

As explained before, once the N-node binary matrix is synthesized, it can be used to solve TSPs 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 tours. This can be expressed mathematically by the following formula:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times [w_{1,2} \quad w_{1,3} \quad w_{2,1} \quad w_{2,3} \quad w_{3,2} \quad w_{3,1}]^T = \begin{bmatrix} w_{1,2} + w_{2,3} + w_{3,1} \\ w_{1,3} + w_{3,2} + w_{2,1} \end{bmatrix}. \quad (7)$$

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 tour. Note that although this example is quite simple, the same method can be carried out for any N-node TSP. After obtaining the length vector, its minimal element coincides with the best tour. As explained before, for other NP-complete problems such as the HPP, there is no need to find the best tour, since any value in the tour-length vector that is equal to N indicates that a Hamiltonian path exists.

3 Optical Implementation

In Sec. 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 implemented optically. 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 with conventional computers, due to the vast amount of memory storage and the complexity of the calculation. Section 3.1 presents the optical implementation of the binary-matrix synthesis according to the binary-matrix algorithm proposed in Sec. 2.1, whereas Section 3.2 presents the optical implementation of the matrix-vector multiplication for obtaining the length vector of the TSP, as explained in Sec. 2.2.

3.1 *Optical Synthesis of the Binary Matrix*

The algorithm proposed in Sec. 2.1 can be used in order to optically synthesize the binary matrix representing all feasible tours of an N-node TSP. In fact, 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

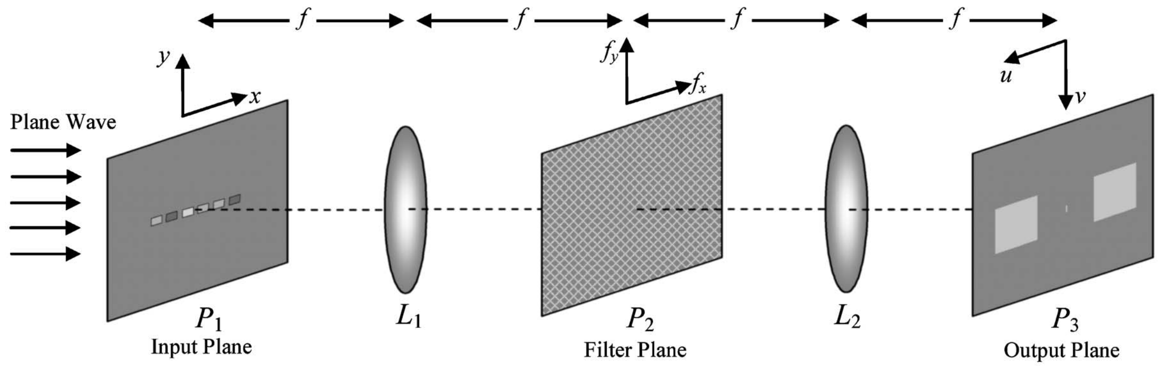


Fig. 4 4f optical system for performing the multiplication of the binary matrix by the weight vector.

duplication is given by Eq. (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[6] 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 vector shifted to the locations of the delta functions. We use pairs of symmetric delta functions in order to get a real-valued spectral transform that can be represented on a regular slide or on a spatial light modulator (SLM).

The binary matrix is also represented on a slide (or an SLM). A white rectangle on this slide represents a 1 in the binary matrix, and a black rectangle on the slide represents a 0 in this matrix. Figure 3 demonstrates the transition from the four-node TSP binary matrix to the five-node TSP binary matrix. As shown in this figure, 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). On the other hand, 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 that 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. After duplicating each of the source-matrix columns by this process, plane P_3 contains the binary matrix of an N-node TSP. Then, 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.

The algorithm described in Sec. 2.1 provides the destination of the duplications for each of the source-matrix columns. This can be accomplished by either the first or the second method, 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 perform that optically, we convolve a pair of shifted symmetric delta functions with the column that should be duplicated. The transformed representation of these two symmetric delta functions can be expressed in the spectral domain as follows:

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

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 that assures the separation of the two matrices that appear on plane P_3.

In the second method, we simultaneously perform a duplication of each of the source-matrix columns into multiple destinations in the target matrix. In order to perform that optically, we convolve a set of shifted symmetric delta-function pairs with the column that is to be duplicated. The transformed representation of this set of shifted symmetric delta-function pairs can be given by the Burch method[7] as

$$\begin{aligned} H^{(2)}(f_x,f_y) = \text{bias} + \Im\Bigg\{ \sum_i [\delta(x-(X_i+A), y-Y_i) \\ + \delta(x+(X_i+A), y+Y_i)] \Bigg\}, \end{aligned} \tag{9}$$

where x and y are the horizontal and vertical axes in the spatial domain, respectively; 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; and $\Im$ is the Fourier transformation operator.

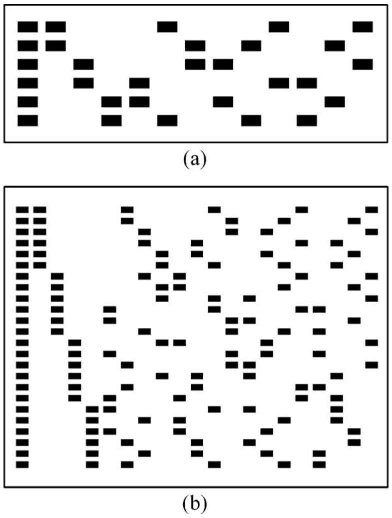


Fig. 5 Simulation results (contrast-inverted pictures): (a) the binary matrix of a four-node TSP (the source matrix); (b) the binary matrix of a five-node TSP (the target matrix).

3.2 *Optical Matrix-Vector Multiplication for Obtaining the Problem Solution*

In order to perform the matrix-vector multiplication, the same 4f optical system used in Sec. 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 grayscale rectangles. The grayscale normalization is performed so that high weights are represented by low grayscale levels.[3] As also shown in Fig. 4, 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,[7] the VanderLugt method,[6,8] 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 product of the binary matrix with the weight vector. This product is the length vector of the TSP. In Ref. 3, we use the joint transform correlator (JTC)[6,9] in order to carry out the multiplication, by using both simulations and lab experiments. In the current paper, we use the 4f optical system to carry out the multiplication. The binary matrix is transformed to the spectral domain by using the Burch method. The implementation of the optical matrix-vector multiplier by the 4f optical system is demonstrated in the current paper by simulations. Simulations and experimental results demonstrating the optical synthesis of the binary matrix, which is explained in Sec. 3.1, are also given in the current paper.

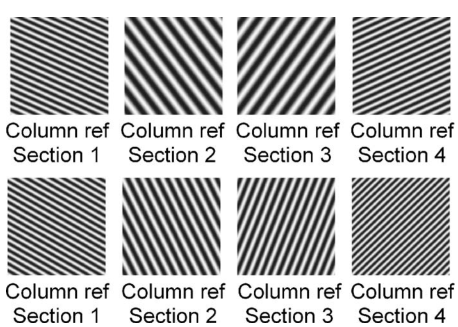


Fig. 6 The first eight out of 56 filters used for the transition from the binary matrix of a four-node TSP to the binary matrix of a five-node TSP in the first method (duplicating into a single location each time). The source-matrix column tag and the target-matrix section number are written below each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

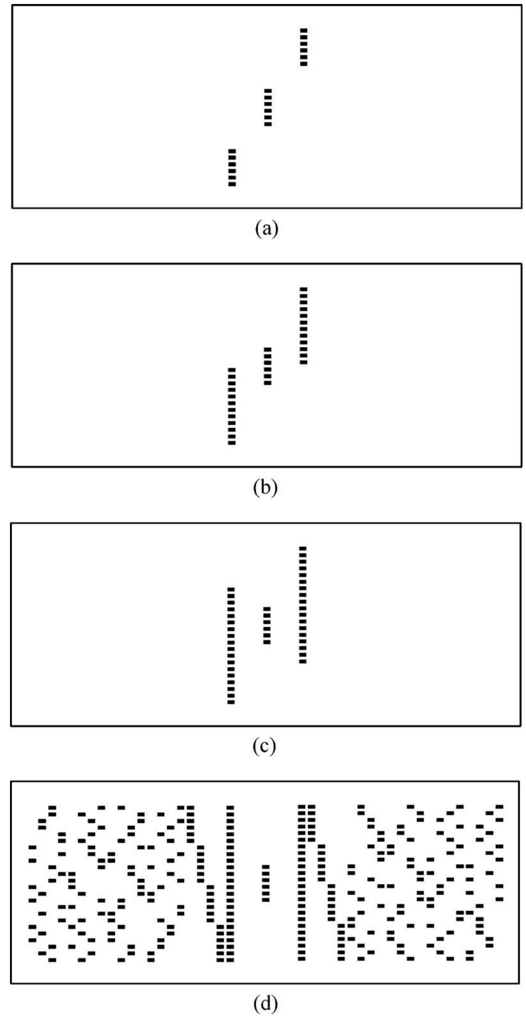


Fig. 7 Four out of 56 contrast-inverted cumulative correlation planes depicting the transition from the binary matrix of a four-node TSP to the binary matrix of a five-node TSP in the first method (duplicating into a single location each time): (a) first duplication; (b) second duplication; (c) third duplication; (d) last (56th) duplication.

4 Simulation Results

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

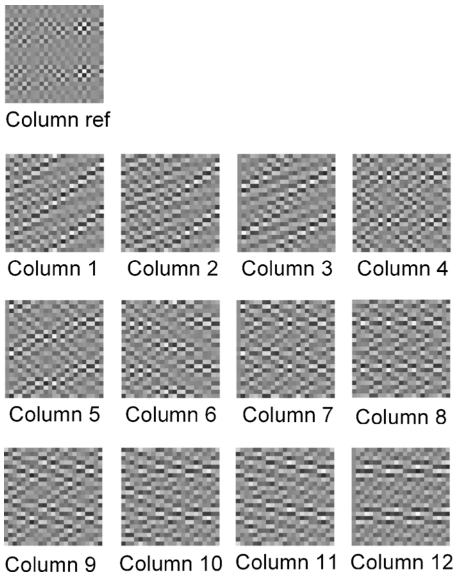


Fig. 8 The 13 filters used for the transition from the binary matrix of a four-node TSP to the binary matrix of a five-node TSP in the second method (duplicating into multiple locations each time). The source-matrix column tag is written below each filter.

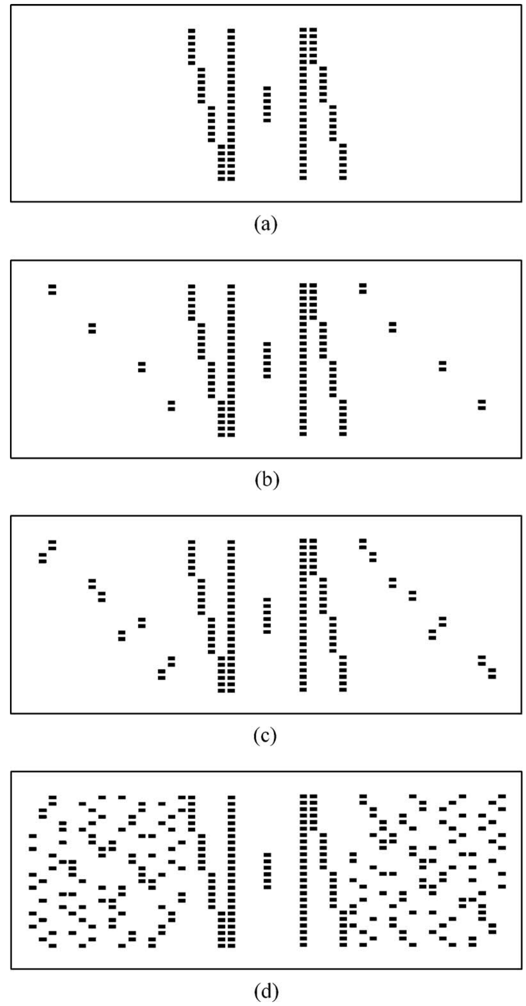


Fig. 9 Four out of 13 contrast-inverted cumulative correlation planes depicting the transition from the binary matrix of a four-node TSP to the binary matrix of a five-node TSP in the second method (duplicating into multiple locations each time): (a) first duplication; (b) second duplication; (c) third duplication; (d) last (13th) duplication.

4.1 *Simulation of the Optical Synthesis of the Binary Matrix*

In order to synthesize the binary matrix of a five-node TSP, we assume the existence of a binary matrix of a four-node TSP (which can be synthesized beforehand from the binary matrix of a three-node TSP by using the same set of rules, defined in Sec. 2.1). To do that, we simulate the 4f optical system illustrated in Fig. 3. On the input plane P_1, we place the binary matrix of a four-node TSP (the source matrix) shown in Fig. 5(a), whereas the cumulative output plane P_3 will eventually contain the binary matrix of a five-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 filter mask of the transformed delta functions.

Simulations of both methods discussed in Sec. 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 source matrix to the target matrix is 56. Thus, 56 different filters [each of which is defined by Eq. (8)] are required in order to perform this transition. Figure 6 shows the first eight out of the 56 filters required for the transition. Figure 7(a)–7(d) show the cumulative output plane P_3 after completing the first, second, third, and 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 Fig. 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. Figure 8 shows these filters. Figure 9(a)–9(d) show the cumulative output plane P_3 after completing the first, second, third, and 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 Fig. 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

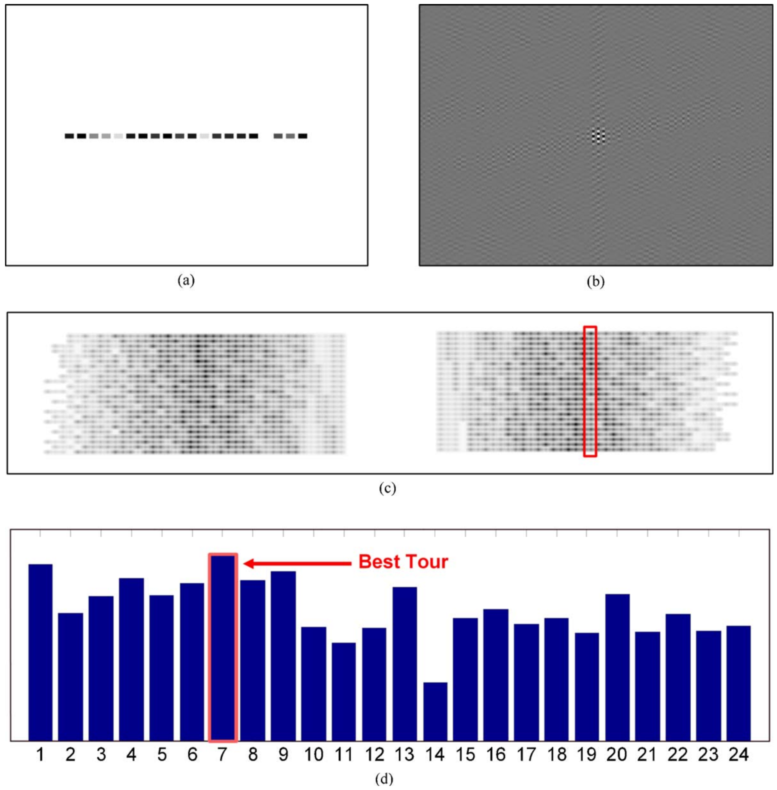


Fig. 10 Simulation results of multiplying the binary matrix by the weight vector in order to obtain the TSP solution: (a) the contrast-inverted weight vector corresponding to the five-node TSP in Fig. 1; (b) the Burch mask of the synthesized binary matrix in Fig. 5(b); (c) the contrast-inverted correlation 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.

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)].

For both methods, the required binary matrix of a five-node TSP (the target matrix) can be easily cut out from the right diffraction order of the output plane P_3 that appears in Fig. 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 to the fact that not all of the columns are symmetric.

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

4.2 *Simulation of the Optical Matrix-Vector Multiplication for Obtaining the Problem Solution*

In this subsection, we demonstrate the solution of the TSP shown in Fig. 1. This is performed (according to the technique explained in Sec. 3.2) by correlating the suitable grayscale weight vector with the binary matrix of a five-node TSP [Fig. 5(b)] that is synthesized in Sec. 3.1. This grayscale 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 tour. This tour is indicated by the strongest spot (or the highest peak) in the middle column of this correlation matrix. The peak heights 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

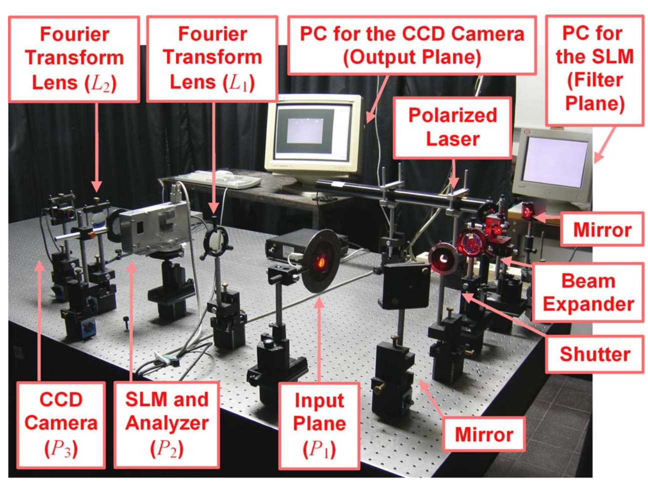
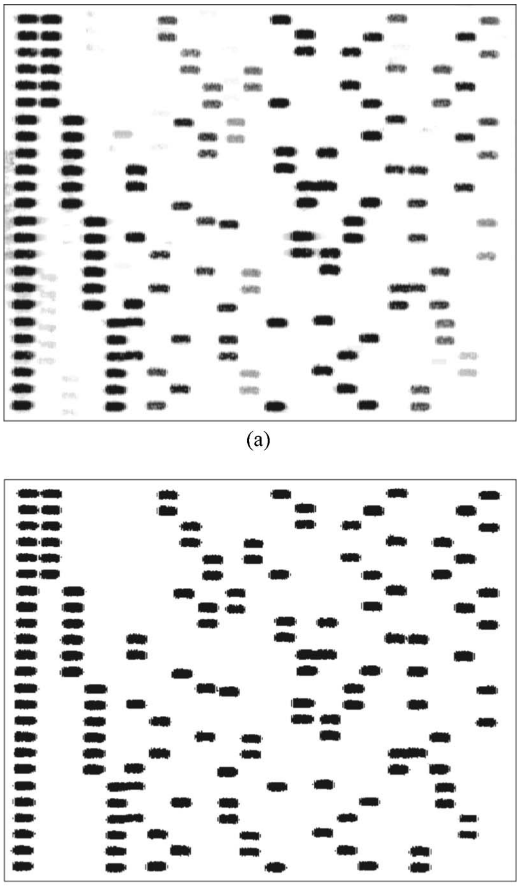


Fig. 11 The optical experiment setup.



(a)

(b)

Fig. 12 The accumulated binary matrix of a five-node TSP obtained experimentally: (a) without the equalization and the thresholding; (b) with the equalization and the thresholding.

seventh tour. This is the shortest tour and thus the solution to the TSP. Going back to the binary matrix [shown in Fig. 5(b)] reveals that this tour 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 tour is node $1 \rightarrow$ node $3 \rightarrow$ node $2 \rightarrow$ node $4 \rightarrow$ node 5 $\rightarrow$ node 1. As can be concluded from Fig. 1, this is indeed the shortest tour.

5 Experimental Results

In this section, we demonstrate by an experiment the synthesis of the binary matrix of a five-node TSP (the target matrix) by using the binary matrix of a four-node TSP (the source matrix). This is performed by implementing the first method (demonstrated by a simulation in Sec. 4.1), in which each column from the source matrix is duplicated into a single location in the target matrix each time. Figure 11 shows a photograph of the experiment setup. As can be seen in this figure, a laser (Uniphase 1144/P, 17 mW, 632.8 nm, 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 25 cm Fourier-transforms the source-matrix column appearing on the input plane, 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). Fifty-six 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 30 cm 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 cumulative output plane, composed of the summation of the 56 resulting correlation planes, is shown in Fig. 12(a). As seen in this figure, this plane indeed contains the target matrix (the binary matrix of a five-node TSP), and it can be compared with the cumulative correlation plane shown in Fig. 7(d), obtained by simulation. Note that since the CCD camera aperture is not large enough to contain the binary matrix of a five-node TSP, we have performed the task by concatenating two CCD camera planes.

The precision of the binary matrix is extremely important, since after the synthesis of this matrix, every row in the matrix is multiplied, element by element, by the weight vector and summed into a single value representing a tour length. Therefore, an improvement of the experimental binary matrix shown in Fig. 12(a) is required. This matrix has two unwanted artifacts, which are mainly caused by the medium quality of the SLM used in the filter plane of the 4f optical system synthesizing the binary matrix. These artifacts are the following: (a) background noise and low-intensity unwanted duplications appear on the cumulative output plane of the optical system; (b) the light intensity on the output plane is not equally distributed along the binary-matrix columns. In order to eliminate artifact (b), one has to equalize the light intensity along the columns. This can be done during the synthesis of the binary matrix by multiplying each duplicated column by a predefined mask placed just in front of the output plane. The transparency values in this mask should turn brighter from its left side to its right side for each concatenated CCD plane in order to compensate for the unequal light intensity values along the binary matrix columns. Artifact (a) (the background noise and the low intensity unwanted duplications) can be eliminated by applying a constant threshold with the CCD camera (or with any other detector used in the output plane, such as an optical film) for each duplicated column during the synthesis of the binary matrix (right after the equalization mask has been applied to this duplicated column).

Figure 12(b) shows the experimental binary matrix after the multiplication by the equalization mask and after applying a constant threshold eliminating the lowest 15% of the values in the output plane during the binary-matrix synthesis. The experimental binary matrix shown in Fig. 12(b) gives only 1.4% erroneous bits, compared to a resized version of the simulated binary matrix shown in Fig. 5(b). Further suppression of the erroneous bits of the experimental binary matrix might be obtained by using more precise optomechanical devices, which are not available in our laboratory at present.

6 Discussion

The experimental results given in the previous section demonstrate a simple proof-of-principle case. However, note that in order to enable a single run (without repetitions) of the proposed method for solving bigger problems with more nodes by using the currently available technologies, the authors think that high-resolution optical films should be employed in the input and output planes of the optical system used for synthesizing the binary matrix. The input film will contain the $(N-1)$-node TSP binary matrix, and on correlating each of this matrix's columns with suitable shifted delta functions, the N-node TSP binary matrix will be accumulated on the output film. After developing the output film, it can be used as the input film of the optical system synthesizing the $(N+1)$-node TSP binary matrix, or alternatively can be used for solving any TSP or HPP containing N or fewer nodes. In spite of the $N-3$ relatively long development processes of the films, needed for obtaining the N-node TSP binary matrix, the required film has to be produced only once for solving any TSP or HPP containing N or fewer nodes, no matter what the problem weights are. Thus, the process of producing such a film can be considered as preprocessing, or as a preproduction of special optical hardware.

The use of a film to represent the binary matrix has another important advantage: the nonlinearities of the film can be exploited during the binary matrix synthesis in order to perform an automatic thresholding process[10,11] on the duplicated columns. As explained in Sec. 5, this thresholding process is important for obtaining a binary matrix with high accuracy.

The authors are aware that currently the proposed optical system cannot compete with electronic computers for TSPs or HPPs of high rank, due to technological limitations. In the future, decreasing the wavelength (which means being able to represent bigger binary matrices) or performing optical iterations with faster SLMs may help if and when the relevant technologies are improved. Therefore, we believe that the proposed optical system is important in its own right.

In addition, even with the currently available technologies, the proposed method has two important features that might make it very useful for many practical applications. These features are the following: (a) the calculation time for obtaining the final solution is defined in advance (since all the feasible solutions are checked and no heuristic or approximation methods are used); (b) after the initial preparation of the binary matrix, the proposed optical system has real-time performance for TSPs and HPPs of low rank (up to 15 nodes). These two features of the optical system can be exploited for cryptography, real-time satellite route decisions, and in general for real-time decision making.

Let us demonstrate the real-time performance of the optical system and compare it with the performance of an electronic computer for a 13-node HPP. In this problem, the number of weights in the (binary) weight vector is $12 \times 13 = 156$, whereas the number of feasible tours is $12! = 4.79 \times 10^8$. Therefore, the number of elements in the binary matrix is $(156+1) \times 4.79 \times 10^8 = 7.52 \times 10^{10}$. Let us assume the use of a Stanford vector-matrix incoherent multiplier.[12] The matrix in this multiplier can be represented on an optical film (slide), whereas the input vector can be represented by vertical-cavity surface-emitting lasers (VCSELs) that can be controlled by a 125-MHz, 9-bit driver. As a result, the dynamic range of the weight vector is 9 bits, which directly affects the accuracy of the optical system so that it cannot solve problems in which the interconnecting weights are represented by more than 9 bits. Under the assumption that the resolution of the binary matrix film is 1 μm per binary matrix element, a 27.4×27.4-cm film can contain the binary matrix of the 13-node problem. After the preproduction of this film, the multiplication of the weight vector, representing the problem weights, by the synthesized binary matrix, representing all feasible tours of the problem, can be performed in the time frame it takes the light to pass through the optical system plus the time it takes to represent the weight vector containing 156 elements by the VCSEL array, which makes up a total of a few nanoseconds. This can be considered as real-time performance. On the other hand, a conventional computer, working in a frequency of a few gigahertz, cannot check 4.79×10^8 tours (without using heuristic or approximation methods) in less than a few tenths of a second, which means that it is 8 orders of magnitude slower than what can be achieved by the proposed optical system, and this cannot be considered as real-time performance.

7 Conclusion

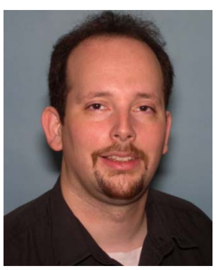
We have proposed an optical method for solving (bounded-length input instances of) NP-complete problems, such as the TSP and the HPP. The method exhaustively checks all feasible solutions of the problem. There is a need to solve this kind of problems by an exhaustive search in order to ensure a predefined solution time. According to the proposed method, we multiply a binary matrix, representing all feasible tours, by a weight vector, representing the weights of the problem. 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 of nodes or fewer. The synthesis of the binary matrix is demonstrated by both computer simulations and an optical experiment. There is good agreement between the simulation and the experimental results. Currently, it is feasible to exhaustively solve TSPs and HPPs which contain 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 only within tens of seconds (cannot be considered as real-time performance). There is still a problem in solving, 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. Decrease of the wavelength, however, is currently quite limited by the currently available light sources and SLMs. Anyway, in our opinion the real-time performance of the system, which can be obtained for small TSPs and HPPs (up to 15 nodes) by using the currently available technologies, testifies to the advantages of the optical system. A possible direction of a future research in this field 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. 20th Annual Conf. on Computational Complexity (CCC)*, IEEE (2005). Longer version available on the Web at http://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," *Appl. Opt.* **46**(5), 711–724 (2007).
4. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed., pp. 282–289, McGraw-Hill, New York (1996).
5. A. D. McAulay, *Optical Computer Architectures: the Application of Optical Concepts to Next Generation Computers*, pp. 289–299, Wiley, New York (1991).
6. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed., pp. 232–246, McGraw-Hill, New York (1996).
7. J. J. Burch, "Computer algorithm for the synthesis of spatial frequency filters," *Proc. IEEE* **55**, 599–601 (1967).
8. A. VanderLugt, "Signal detection by complex spatial filtering," *IEEE Trans. Inf. Theory* **IT-10**, 139–145 (1964).
9. C. S. Weaver and J. W. Goodman, "A technique for optically convolving two functions," *Appl. Opt.* **5**, 1248–1249 (1966).
10. D. G. Feitelson, *Optical Computing: A Survey for Computer Scientists*, p. 98, MIT Press, Cambridge, MA (1988).
11. N. H. Farhat, "Nonlinear optical data processing and filtering: a feasibility study," *IEEE Trans. Comput.* **24**(4), 443–448 (1975).
12. J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed., pp. 284–286, McGraw-Hill, New York (1996).



Natan T. Shaked received his BSc and MSc (summa cum laude) degrees in electrical and computer engineering from Ben-Gurion University of the Negev, Beer Sheva, Israel, in 2002 and 2004, respectively. Currently, he is working on his PhD in electrical and computer engineering and is an instructor in the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev. He has coauthored six scientific papers in refereed journals and presented his work at seven different academic conferences. His current research interests include incoherent white-light holography and optical computing and processing.

Tal Tabib received his BSc degree in electrical and computer engineering from Ben-Gurion University of the Negev in 2006. In his BSc, he majored in computers and communication systems, and performed final BSc project in optics (contributed to this paper). Currently, he is working as a logic design engineer at a leading communication company providing unique wireless LAN solutions.

Gil Simon received his BSc degree in electrical and computer engineering from Ben-Gurion University of the Negev in 2006. In his BSc, he majored in electro-optics, computer networking, and digital signal processing, and performed BSc project in optics (contributed to this paper). Currently, he is working toward his MSc in digital signal processing at Tel-Aviv University, Israel, and is a hardware engineer at a leading company providing highly integrated silicon solutions.

Stephane Messika has been a computer science assistant professor at Orsay Paris-Sud University since 2005. After receiving his MSc degree in mathematics, he completed his PhD degree at École Normale Supérieure (ENS) Cachan, France in 2004. During his postdoctoral studies at Ben-Gurion University of the Negev, he worked on optical solutions of instances of NP-complete problems. His research interests include probabilistic algorithms, model checking, and formal verification. He received the Best Student Paper prize at the DISC 2004 conference for his work on coupling for probabilistic algorithms.

Shlomi Dolev received his BSc degree in engineering and BA degree in computer science in 1984 and 1985, respectively, and his MSc and DSc degrees in computer science in 1990 and 1992, respectively, from the Technion-Israel Institute of Technology. He is the author of the book *Self-Stabilization*, published by the MIT Press in 2000, and is an associate editor of *IEEE Transactions on Computers* and of the AIAA *Journal of Aerospace Computing, Information, and Communication*. He established the Computer Science Department at Ben-Gurion University of the Negev, and served as the first chair of this department. He is now a professor and holds the Rita Altura Trust Chair in Computer Sciences. His current research interests include distributed computing, distributed systems, communication networks, self-stabilization systems, cryptography, and optical computing.

Joseph Rosen received his BSc, MSc, and DSc degrees in electrical engineering from the Technion-Israel Institute of Technology in 1984, 1987, and 1992, respectively. He is currently a professor in the Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev. He is a fellow of the Optical Society of America and has coauthored more than 70 scientific papers in refereed journals. His research interests include image processing, holography, diffractive optics, interferometry, pattern recognition, optical computing, and statistical optics.