Editors' Suggestion

Scalable and programmable three-dimensional photonic processor

Xi Tan,[1,2,3] Hang Song,[1,3] Ye Ji,[4] Hao Tang,[1,3] Ying-Yu Fang,[1,3] Xiao-Yun Xu,[1,3] Yang-Yang Li,[4] Xuan-Kun Li,[1,3] Ka-Di Zhu,[2,*] and Xian-Min Jin[1,3,4,5,†]

[1] *Center for Integrated Quantum Information Technologies (IQIT), School of Physics and Astronomy and State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China*

[2] *Key Laboratory of Artificial Structures and Quantum Control (Ministry of Education) and School of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai 200240, China*

[3] *Hefei National Laboratory, Hefei 230088, China*

[4] *TuringQ Co., Ltd., Shanghai 200240, China*

[5] *Chip Hub for Integrated Photonics Xplore (CHIPX), Shanghai Jiao Tong University, Wuxi 214000, China*

(Received 9 March 2023; revised 22 August 2023; accepted 1 September 2023; published 16 October 2023)

The maximum clique problem (MCP) is a typical nondeterministic polynomial-time complete (NP-complete) problem that is considered to be intractable for classical computers. Photons are expected to tackle these computationally hard problems as they are intrinsically suitable for parallel computing. Here, we propose and experimentally demonstrate a scalable and programmable three-dimensional photonic processor that can efficiently solve the MCP with photons in a parallel manner. By designing three-dimensional waveguide networks, we map the MCP into a photonic chip fabricated by femtosecond laser direct writing. The prototype demonstrates the superiority of photonic parallel computing based on non-von Neumann structures. Furthermore, we show that our photonic processor can be applied to search for the optimal molecular docking configurations, a major issue in pharmaceutical medication development. We manage to map different graphs into a single three-dimensional waveguide network by encoding injection conditions of photons, which build an alternative paradigm to indirectly program three-dimensional photonic chips. The combination of parallel and fully programmable capabilities of photonic computing paves the way for wider applications.

DOI: 10.1103/PhysRevApplied.20.044041

I. INTRODUCTION

In graph theory, a simple, finite, and undirected graph $G = (V, E)$ is defined as a set containing vertices and edges. A clique C is a fully connected graph in the subgraphs of G, $(C \in G)$ [1]. Given an arbitrary graph, the maximum clique problem (MCP) investigates which clique contains the most vertices. In recent years, the MCP has been extensively studied for its wide applications in sociology, biology, mathematics, information science, etc. [2]. Generally, to solve this problem, the unsophisticated and, for most of the time, the best way is to enumerate all possible subgraphs, and then return the maximum-sized clique. Obviously, for electronic computers, the computational resources' consumption required to solve this problem increases exponentially with the problem size [3]. These kinds of problems, such as the traveling salesman problem [4,5], the vertex cover problem [6,7], the subset sum problem [8,9], etc., are also called nondeterministic polynomial-time complete (NP-complete) problems [10]. It remains an open question whether there are polynomial-time algorithms for the NP-complete problems, namely, whether $P = NP$ [11].

In view of the difficulty of solving the NP-complete problem lies in traversing all possible candidate solutions. This motivates us to develop parallel computing frameworks to meet this challenge. However, the von Neumann framework [12], which has been widely used in the design of modern electronic computers, separate memory and computating units, and execute computations sequentially, making parallel computing problematic [13]. Fortunately, some researchers are committed to the development of computing paradigms to solve these intractable computational problems, for instance, quantum computing [14–17], photonic computing [9,10], DNA computing [18–20], microfluidic computing [3,21], etc. (see Appendix A). Among these schemes, photons stand out for their many extraordinary features suitable for parallel computing, such as fast propagation speed for high-speed computing [22–24], multiple degrees of freedom (DOFs) for

*zhukadi@sjtu.edu.cn
†xianmin.jin@sjtu.edu.cn

encoding [25,26] and strong robustness against tunneling and losses [27].

In this paper, we present a scalable and fully programmable non-von Neumann computational framework based on photon computing for solving the MCP in a parallel computing fashion. Similar to how electrical signals behave on classical computers, here we use photons as information carriers to implement computing tasks parallelly in the photonic chip. Generally, most of the current two-dimensional photonic chips rely on embedded thermo-optical phase shifters (TOPSs) [26,28,29] or electro-optical modulators (EOPMs) [26,30] for programming, which is hardly generalizable to three-dimensional structures. Skillfully, we adopt a different approach to programming by controlling the injection condition of coded photons, which creates an alternative paradigm for implementing programming in three-dimensional photonic chips. With the full programmability, our six-vertex-customized-chip processor can solve the MCP for up to 32 767 different graphs in experiment (see Appendix E). Moreover, our scheme exhibits significant superiority in time consumption over electronic computers. Furthermore, we study the utility of our processors in solving molecular docking problems, demonstrating the potential of photonic computing in solving real-world problems.

II. EXPERIMENT

Our scheme provides a three-dimensional photonic chip processor as an optical oracle for the solution of the MCP. As shown in Fig. 1(a), the processor consists of three parts: programming unit, processing unit, and read-out unit. The programming unit mainly includes a field programmable gate array (FPGA) and a series of lasers. Its main function is to encode the injection conditions of photons by controlling the switches of different channel lasers. The processing unit, the core of our processor, consists of a three-dimensional photonic chip fabricated by femtosecond laser direct writing (see Appendix B for more details). The design of the chip is inspired by the microfluidic computational scheme proposed by Whitesides *et al.* [3]. The read-out unit chiefly contains a CCD to record and output the calculation results. The process and principle of our processor for solving the MCP will be explained in detail below.

The photonic chip is the "brain" of our processor resembling a central processing unit (CPU) to the electronic computer. Inside the chip are customized three-dimensional waveguide computing networks. Taking a five-vertex fully connected graph $\{ABCDE\}$ displayed in Fig. 1(b) as an example, its corresponding three-dimensional waveguide network structures inside the chip are illustrated in Fig. 1(c). The front of the waveguide networks are parallel, equally spaced input waveguides buried in the chip. Each input waveguide in Fig. 1(c) represents one of the possible edges in Fig. 1(b) in turn. Then, each input waveguide successively connects to a single-layer waveguide network of different depth in the chip by bending along the three directions of the *XYZ* axis. In each single-layer waveguide network, the input waveguide is split into multiple output waveguides by beam-splitting structures [Fig. 1(e)]. While the input waveguide represents an edge, the output waveguides in feature positions represent all possible subgraphs that contain this edge. For example, as presented in Fig. 1(d), the input waveguide representing the edge $[CE]$ is successively split into eight waveguides through the beam-splitting structures [Fig. 1(e)], and then all of these waveguides extend to the feature positions represented by the subgraphs that contain the edge $[CE]$. Other layers' waveguide structures are analogous to that of the *CE* layer. In the *YZ* plane, the distribution of waveguide output ports in the chip is demonstrated in Fig. 1(f).

The key to the well functioning of the photonic chip relies on the beam-splitting structure, which is a deformed waveguide directional coupler. As shown in Fig. 1(e), it consists of an input waveguide and a coupled waveguide. According to the coupled-mode theory (CMT) [31–33], the splitting ratio can be customized by modulating the coupling distance (*cd*) or coupling length (*cl*). In our experiment, we fix the coupling distances to 10 μm, and then modulate the coupling lengths to realize the desired beam-splitting ratio. Once the coupling waveguide of the previous node extends to the next beam-splitting structure, it automatically transforms to the input waveguide.

In our scheme, the process of solving the MCP can be roughly divided into four steps: (i) We start by sorting out all connected edges of the given graph, then convert them into binary codes and send them to the FPGA as input signals. (ii) Then the FPGA controls the switching of the different lasers according to the received commands. Thus, the input information is encoded into the photons of different channels. (iii) Subsequently, the encoded photons are coupled to different input ports of the photonic chip via a V-groove fiber holder. Afterwards, the photons are guided in parallel by the waveguide networks to feature positions at the end of the chip. (iv) Eventually, the photons carrying the calculation results are emitted from the chip and recorded by the CCD. The maximum clique can be inferred from the spot image according to the following judgment rule (see Appendix C for more details).

The principle of the judgment rule [3] is as follows: for a candidate subgraph with m vertices, if it can form a clique, then any two of its vertices must be connected, that is, it must contain $\binom{m}{2} = m(m-1)/2$ edges. Under this criterion, all the cliques of the given graph can be identified from the spot image. Moreover, the photons are transmitted independently and parallelly in the chip. Therefore, the parallel computing fashion has been created in our processor.

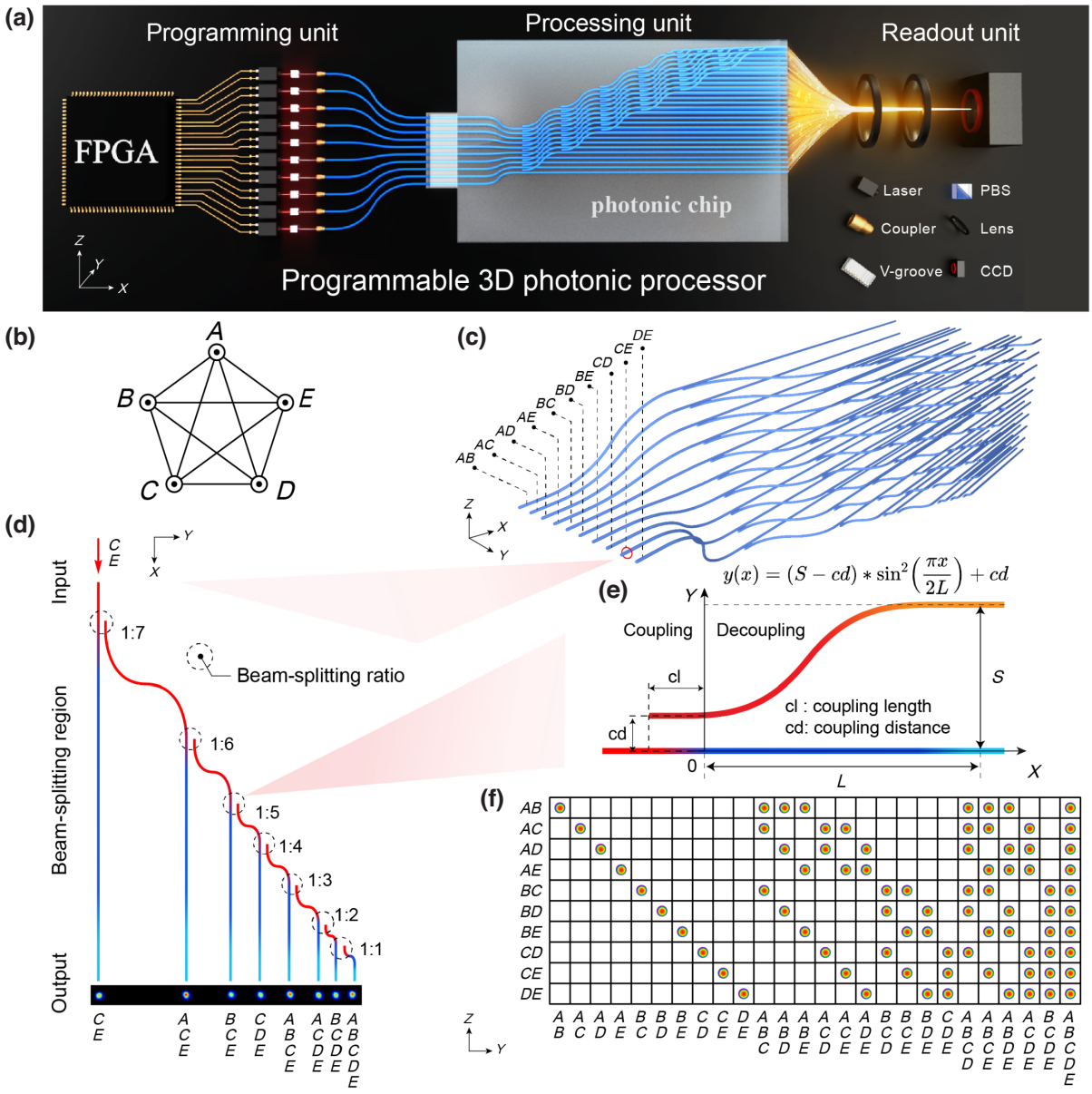

FIG. 1. Architecture of the photonic processor. (a) Total view of the programmable three-dimensional photonic chip processor. (b) A specific case of a five-vertex graph $\{ABCDE\}$. (c) Schematic diagram of the three-dimensional waveguide structure inside the photonic chip for solving the MCP of the five-vertex graph. (d) The single-layer waveguide network structure marked $[CE]$. The ideal beam-splitting ratios displayed in the figure ensure near-perfect equal intensity output from the waveguide port. (e) Partially enlarged structural diagram of the beam-splitting structure. In the coupling region, the input and coupling waveguides remain parallel. In the decoupling region, the coupled waveguide is decoupled along the equation curve of the form $y(x) = (S - cd) * \sin^2\left(\frac{\pi x}{2L}\right) + cd$, where S represents the actual spacing between the two waveguides. (f) Cross-section view of the output waveguide distribution at the end face of the chip.

III. RESULTS

A. Experimental demonstrations

We demonstrate two experimental results of solving the MCP for a five-vertex graph and a six-vertex graph with our programmable photonic processor. As exhibited in Figs. 2(a) and 2(b), each case shows a randomly given graph to be solved, the corresponding experimental readout, and the calculated results of the maximum clique. The spot image on the dark blue background is the experimental readout recorded by the CCD. Below the spot image are blue bars of the sum of the number of spots in each column (feature position) accompanied by red bars of the judgment thresholds.

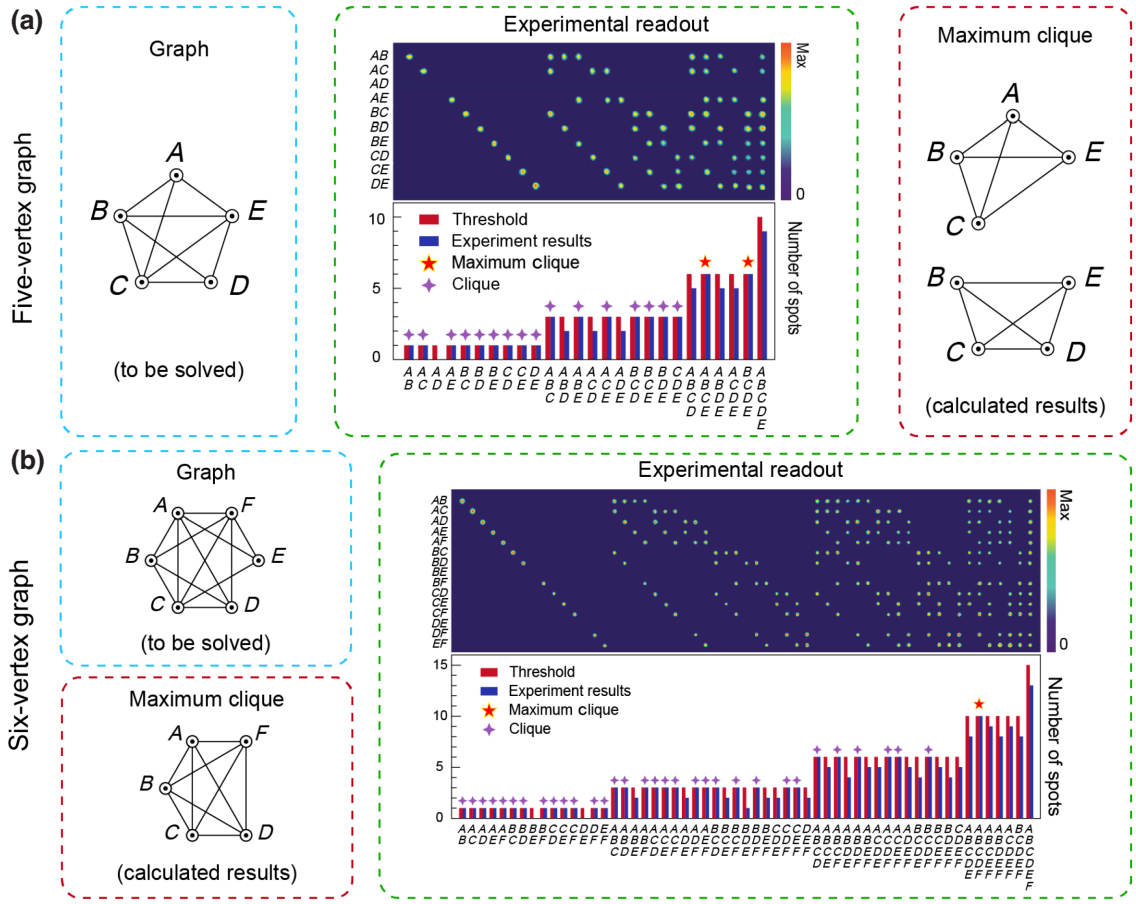

FIG. 2. Experimental readout and calculation results. (a),(b) are cases of using our photonic processors to solve the MCP for randomly generated graphs, respectively. The spot image is the experimental readout recorded by the CCD. The blue and red bars at the bottom of the image indicate the sum of the number of spots in each column, and the corresponding judgment threshold, respectively. Through contrasting the blue bars and red bars, all the cliques can be determined.

By sampling and averaging the background noise, we make a standard value to determine whether there is a signal output at the exit port of each waveguide. Due to the nonideal beam-splitting ratio caused by imperfections in chip fabrication, the intensity of different spots is not exactly equal as the ideal distribution, as seen in the experiment readout in Figs. 2(a) and 2(b). However, what we actually care about is the intensity of the spot signal relative to the background noise, and our calculations are valid as long as the weakest spot exceeds the standard value. Furthermore, we simulate the effect of nonideal splitting ratios on the output power of the waveguide, and the results show that a relatively stable distribution can be maintained by compensating for systematic errors within the experimentally allowed range (see Appendix D). Aside from the aforementioned cases, we also perform extensive experimental validation by solving the MCP for different graphs (see Appendix E).

B. Time-consumption performance

To evaluate the time-consumption performance of our photonic processor, we compare it with up-to-date CPU [34], GPU [35], and supercomputer [36], respectively. Since our processor can not only obtain the maximum clique, but also output all subgraph information of the given graph simultaneously in one operation. On the premise of obtaining all subgraph information, our photonic processor can be extended to tackle a series of maximum clique-based variant NP-complete problems, such as the maximum-weighted clique problem [37,38], the quasiclique problem [39,40], the top-k densest-subgraph problem [41,42], etc. Therefore, to comprehensively evaluate the time-consumption performance of our photonic processor, we compare the computational time required for electronic computers to enumerate all subgraphs for the same problem size (see Appendix F for details). As shown in Fig. 3(a), the photonic processor are predicted to

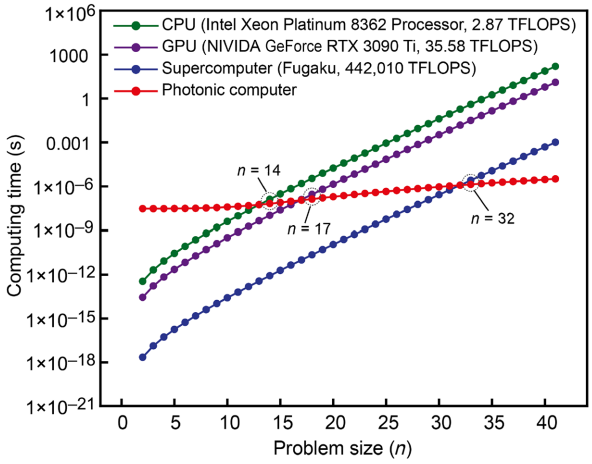

FIG. 3. Time-consuming performance. Comparing the computing time of the photonic processor and up-to-date electronic computers for the same computation task. CPU, GPU, and supercomputer are surpassed at $n = 14$, $n = 17$, and $n = 32$, respectively.

outperform CPU, GPU, and supercomputing at the size of 14, 17, and 32, respectively.

C. Solving the molecular docking problems

Further we demonstrate the potential of our photonic processing in solving practical problems by solving a molecular docking case. Molecular docking is a useful method for drug design [43]. Usually, different binding poses between a macromolecular protein (receptor) and a molecule drug (ligand) will form different three-dimensional spatial conformations. The main purpose of molecular docking is to find the optimal docking pose to determine the structure of the final stable complex [44]. To denote the key sites of the molecule, we use pharmacophores [45,46] to represent the active chemical characteristic structures and key interaction centers in the molecule (see Appendix G). We define the following six different types of key pharmacophores [46]: hydrogen-bond donor, hydrogen-bond acceptor, positive charge center, negative charge center, hydrophobe, and aromatic ring center. For convenience, we employ a labeled distance graph to replace the all-atom model of the actual molecule [46]. As presented in Fig. 4(a), taking the small molecule VWW [47] as an example, the processes of representing a receptor or ligand molecule as a labeled distance graph are as follows:

(1) Identify all the pharmacophores on the molecule that may be involved in the binding action, and distinguish them with different labels and colors [Fig. 4(b)].

(2) Calculate the Euclidean distances between the different pharmacophores on the molecule [Fig. 4(c)].

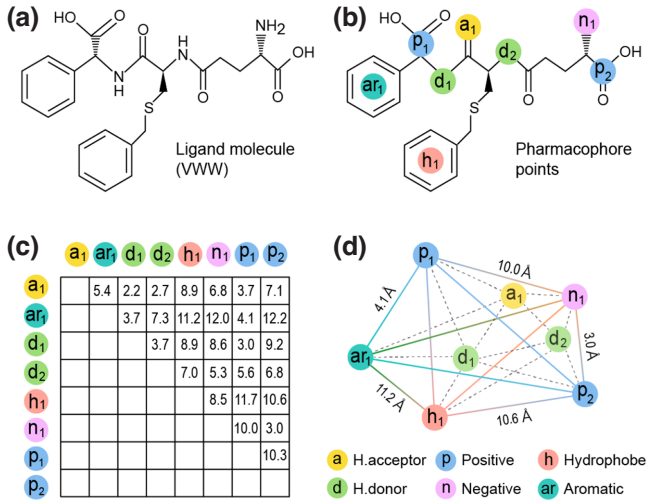

FIG. 4. Construction of the labeled distance graph for a molecule. (a) Planar structure diagram of the molecule (VWW). (b) The pharmacophore points in the molecule are identified and distinguished with different colors and labels. (c) Calculate the Euclidean distance between the pharmacophore pairs (unit: Å). (d) Construction of a labeled distance graph instead of the all-atom model of the molecule.

(3) Construct a fully connected graph, where the vertices of the graph represent the pharmacophore on the molecule, and the labeled values on the edges denote the Euclidean distance between the pharmacophores on the actual molecule [Fig. 4(d)].

Adopting the above method, we construct the labeled distance graphs G_R and G_L of the macromolecule protein receptor 3MZ3 [47] and the small-molecule ligand B3N [47], respectively, to study their binding interactions, as depicted in Fig. 5(a). In the process of molecular docking, the pharmacophore points on the receptor 3MZ3 and the ligand B3N can contact with each other to produce different combinations of pharmacophore pairs, as shown in Fig. 5(b). In principle, for a receptor with m pharmacophores and a ligand with n pharmacophores, there are mn possible free combinations [46]. However, not every combination formed by contact satisfies the chemical complementarity and spatial geometry matching rules (see Appendix H). We first consider the following rules for chemical complementarity [45,48] between pharmacophore points, as displayed in Fig. 5(c):

(1) Hydrogen-bond interactions: hydrogen-bond donors correspond to hydrogen-bond acceptors.
(2) Electrostatic interactions: positive charge centers correspond to negative charge centers.
(3) Hydrophobic interactions: hydrophobic centers correspond to the hydrophobic centers.
(4) $\pi - \pi$ interactions: aromatic ring centers correspond to aromatic ring centers.

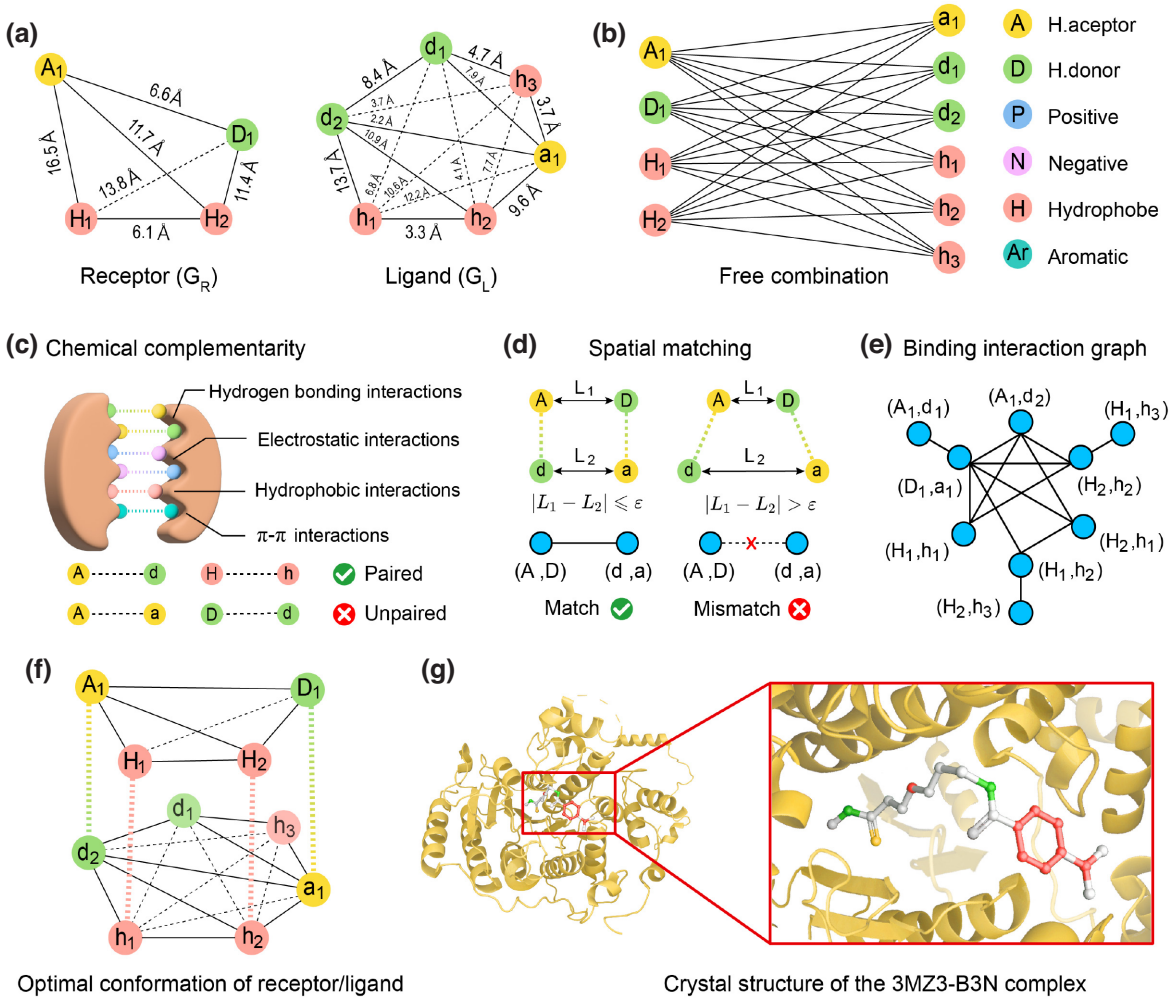

FIG. 5. Mapping the molecular docking problem to the MCP. (a) Construction of labeled distance graphs for receptor (3MZ3) and ligand (B3N). We denote the pharmacophores on the receptor and ligand with uppercase and lowercase letters, respectively. (b) All possible combinations of pharmacophores under free binding. (c) Principles of chemical matching of pharmacophore pairs. (d) Principle of spatial matching between two chemically matched pharmacophore pairs. (e) Construction of a binding interaction graph for receptor (3MZ3) and ligand (B3N). (f) The spatial conformation between the receptor and ligand labeled distance graphs is inversely deduced from the results of the maximum clique on the binding interaction graph. (g) The crystal structure of the stable complex between the receptor (3MZ3) and ligand (B3N) molecules is obtained according to the docking configuration in (f).

After screening by the chemical complementarity rules, the pairable pharmacophore pairs were retained, and the unpaired cases were excluded (see Appendix I). However, the chemically complementary pharmacophore pairs are not necessarily physically compatible with each other [46, 49]. Therefore, we consider the rules of spatial matching between pharmacophore pairs. As illustrated in Fig. 5(d), for the two chemically complementary pharmacophore pairs (A, d) and (D, a) from receptor and ligand, we judge whether they are physically compatible by comparing their Euclidean distances $L_1 = |AD|$ and $L_2 = |da|$. In general, the spatial distance of the two groups of pharmacophore points that can coexist on the receptor and ligand should be substantially equivalent, but with some flexibility. Therefore, if $|L_1 - L_2| \leq \varepsilon$, the two sets of pharmacophore pairs (A, d) and (D, a) are regarded as satisfying the spatial matching conditions [46], and vice versa (see Appendix I). Here, ε characterizes the spatial matching flexibility in the binding process, which is generally based on empirical knowledge acquisition [46]. In our experiments, we set ε to 4.4 Å.

To be more intuitive, we introduced a binding interaction graph to represent the chemical complementarity and spatial matching information between receptor and ligand, as shown in Fig. 5(e). The rules for the construction of the binding interaction graph are as follows:

(1) Any vertex on the binding interaction graph is a set of chemically matched pharmacophore pairs from ligand and receptor, respectively.

(2) Any two vertices on the binding interaction graph are connected by an edge if they satisfy the spatial matching condition, that is, they can coexist.

In this way, we transfer the information of compatible pharmacophore pairs on the 3MZ3 and B3N to the binding interaction graph [Fig. 5(e)]. In order to search for the most stable three-dimensional configuration of the 3MZ3-B3N complex in molecular docking, we ought to find the largest group of coexisting pharmacophore pairs. This is namely to find the maximum clique on the binding interaction graph. Thus, we have successfully mapped the molecular docking problem to a maximum clique problem, which can then be solved with our photonic processor. Subsequently, according to the maximum clique $\{(A_1, d_2), (H_1, h_1), (H_2, h_2), (D_1, a_1)\}$ in the binding interaction graph, we reconstruct the optimal spatial conformation of receptor and ligand with correct pharmacophore interactions, as exhibited in Fig. 5(f). Finally, the crystallographic structure of the 3MZ3-B3N stable complex can be predicted from the docking configuration, as the enlarged view reveals in Fig. 5(g).

IV. CONCLUSION AND DISCUSSION

In summary, we have proposed and built a prototype of a scalable and fully programmable parallel computing photonic processor to address the MCP. Our coding scheme to control the photon injection conditions fills the gap in realization of programming three-dimensional photonic chips. The utility of our processor has been demonstrated in solving molecular docking problems. Benefiting from non-von Neumann computational frameworks and photonic parallelism, our results suggest that for moderate problem sizes, the time-consuming performance of the photonic processor has the potential to surpass supercomputers towards "photonic advantage".

As one of the NP-complete problems, the MCP can be efficiently solved by our photonic processor. In view of the fact that all NP-complete problems can be reduced to each other in polynomial time [50], it may imply that there will be more intractable problems that can be mapped to our photonic chips to solve. This will also greatly promote the prosperity and development of photonic computing. Moreover, the photons have various DOFs suitable for encoding [25,26]. Thus, in the future, the parallel computing capability of photons can be enhanced by combining different DOFs, such as frequency [30,32], polarization [28,31], orbital angular momentum [25], which will enable our photonic processors to be further accelerated.

ACKNOWLEDGMENTS

This research is supported by the National Key R&D Program of China (Grants No. 2019YFA0308700, No. 2019YFA0706302, and No. 2017YFA0303700); National Natural Science Foundation of China (NSFC) (Grant No. 62235012, No. 11904299, No. 61734005, No. 11761141014, No. 11690033, and No. 12104299); Innovation Program for Quantum Science and Technology (Grants No. 2021ZD0301500, and No. 2021ZD0300700); Natural Science Foundation of Shanghai (Grant No. 20ZR1429900); Science and Technology Commission of Shanghai Municipality (STCSM) (Grants No. 20JC1416300, and No. 2019SHZDZX01); Shanghai Municipal Education Commission (SMEC) (Grant No. 2017-01-07-00-02-E00049); China Postdoctoral Science Foundation (Grants No. 2021M692094, and No. 2022T150415). X.-M.J. acknowledges additional support from a Shanghai talent program and support from Zhiyuan Innovative Research Center of Shanghai Jiao Tong University.

APPENDIX A: COMPUTATIONAL BOTTLENECKS AND NON-VON NEUMANN STRUCTURES

The invention of electronic computers has greatly promoted the development of science and the progress of human society. Nowadays, the application of an electronic computer has penetrated into many aspects of life. However, with the continuous development of science and technology, human society has produced an unprecedented amount of information and data. It also makes us more dependent on the computing power of electronic computers than at any time in history. Since the birth of the world's first electronic computer, mankind's pursuit of computing power has never stopped. In the past few decades, scientists have been working to increase the integration of chips to radically increase the computing power of computers. The development of chips fits well with Moore's law [51]. However, with the reduction of the chip manufacturing process, the current chip manufacturing also faces huge challenges such as processing technology and transistor heat dissipation [52]. The development trend of chips has gradually become slow. As the transistor process continues to decrease, the quantum tunneling effect [53] will become non-negligible at this time, and the transistor will become no longer reliable, which is also known as the "Moore's law bottleneck" [54].

Most modern electronic computers are based on the von Neumann architecture [12], which separates the memory from the central processor and performs computations in a sequential manner, as shown in Fig. 6. This approach has improved versatility, enabled programmable computing functions, and greatly facilitated the development of computers. However, this structure also creates a potential problem for the development of the computer. Since the computing module and the storage module are separated, this causes the central processing unit to be in an idle state of waiting when data is written to and read from the

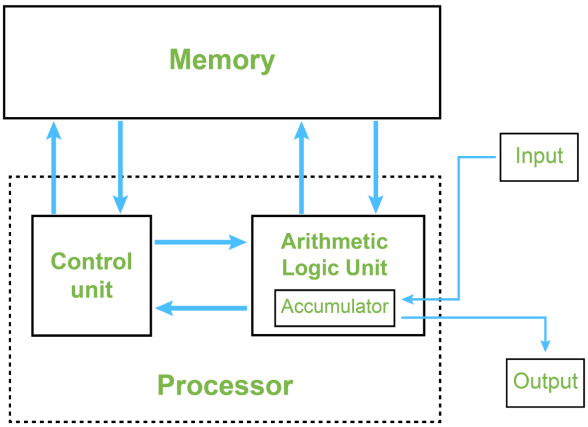

FIG. 6. Schematic diagram of the von Neumann computer architecture.

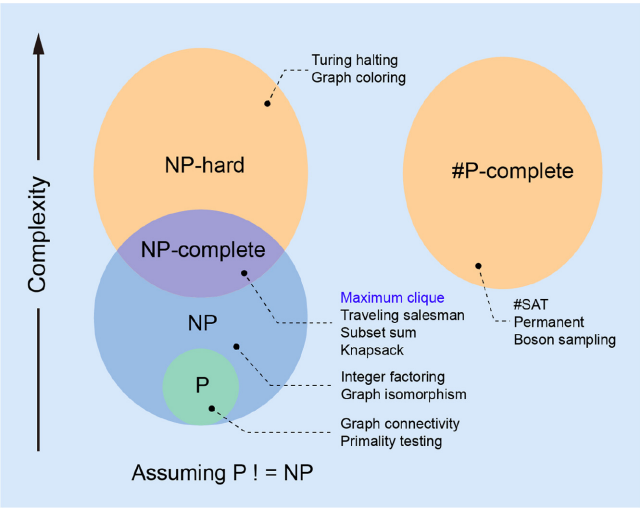

FIG. 7. Computational complexity class of the problems. Assuming P != NP, the computational complexity of P problems, NP problems, NP-complete problems, NP-hard problems, and #P-complete problems, and corresponding problem instances.

memory. However, today, the processing speed of the processor has far exceeded the access speed of the memory, so the improvement of the performance of the processor is very limited to the improvement of the overall computing efficiency, which is the so-called "von Neumann bottleneck" [55]. Electronic computers based on the von Neumann structure are good at repeating operations in order of instructions, so they can solve P problems [56] perfectly (problems that can be solved in polynomial time). However, as demonstrated in Fig. 7, for problems such as the NP-complete problem [57], NP-hard problem [58], where the computational effort grows exponentially with the size of the problem, the electronic computers cannot complete the computation in a reasonable time. Throughout the development of electronic computers, "Moore's law bottleneck" and "von Neumann bottleneck" are the two bottlenecks that strongly limit the development of computer computing power. Therefore, there is an urgent need to develop alternative computing paradigms to meet human demand for computing power.

Taking inspiration from nature, we find that the secret of the high-speed operation of neural networks in the brain lies in the high parallelism between neurons, which indicates that the computing speed can be increased rapidly by developing parallel operations. However, most modern electronic computer systems are based on the von Neumann sequential control flow structure, which fundamentally limits parallelism. Therefore, in order to realize parallel computing, some non-von Neumann-based computing frameworks have been proposed, such as photonic computing [9,10], quantum computing [14–16,46,59–65], DNA computing [18–20,66–68], microfluidic computing [3,21], biomolecular computing [8,69–71], etc. As presented in Table I, non-von Neumann parallel computing schemes based on different systems have been continuously proposed and improved, and are developing vigorously.

As a promising candidate, photons have natural parallel capabilities and are suitable as carriers for the development of high-speed and low-energy parallel computing. In our scheme, we employ photons as the information carrier, and use a special photonic chip to build a photonic processor with a non-von Neumann computing architecture, and successfully solve the maximum clique problem. Different from the serial working mode of electronic computers, photons perform computing tasks in parallel in our photonic chip, which makes our scheme more advantageous in time consumption.

APPENDIX B: PHOTONIC CHIP FABRICATION AND EXPERIMENTAL SETUP

The three-dimensional waveguide networks inside the photonic chip are prepared by the femtosecond laser with a central wavelength of 513 nm, a repetition rate of 1 MHz, a pulse duration of 290 fs. We used a spatial light modulator (SLM) to shape the laser pulse and then focused it inside the borosilicate glass (Eagle XG) substrate through a 50× objective lens (numerical aperture, 0.55). The high-precision ABL1500 series three-dimensional air-bearing translation stage guarantees that the three-dimensional waveguide can be accurately processed inside the medium. In our experiments, the translation stage was moved at a speed of 10 mm/s to prepare the waveguide. The power compensation mechanism ensures that the waveguides are uniform and smooth at different depths [72]. The built-in active power-lock feedback device of the direct writing system maintains the long-term stability and repeatability of processing parameters.

When solving the maximum clique problem, the most crucial component is to construct a customized photonic chip with special computing networks in line with the number of vertices of the given graph. The input

TABLE I. Various non-von Neumann computational frameworks based on different systems have been used to solve hard problems that are intractable with classical von Neumann structures.

Types	Research issues	Complexity classes	Performance		Representative works
			Scalability	Programmability	
Photonic computing	Maximum clique problem	NP-complete	✓	✓	Our work ★
	Subset sum problem	NP-complete	✓	✗	Ref. [9]
	Hamiltonian path problem	NP-complete	✗	✗	Ref. [10]
Quantum computing	Gaussian boson sampling	#P-complete	✓	✓	Ref. [16]
	Maximum independent set	NP-complete	✗	✓	Ref. [15]
	MaxCut problem	NP-complete	✓	✓	Ref. [59]
	MaxCut problem	NP-complete	✗	✓	Ref. [60]
	Graph isomorphisms	NP	✓	✓	Ref. [61]
	Gaussian boson sampling	#P-complete	✓	✗	Ref. [14]
	Maximum weighted clique	NP-complete	✗	✓	Ref. [46]
	Boson sampling	#P-complete	✓	✗	Ref. [62]
	Boson sampling	#P-complete	✓	✗	Ref. [63]
	Boson sampling	#P-complete	✓	✓	Ref. [64]
	Boson sampling	#P-complete	✓	✗	Ref. [65]
DNA computing	Assignment problem	NP-hard	✗	✓	Ref. [20]
	SAT problem	NP-complete	✗	✓	Ref. [19]
	SAT problem	NP-complete	✗	✓	Ref. [66]
	SAT problem	NP-complete	✗	✓	Ref. [18]
	Maximum clique problem	NP-complete	✗	✓	Ref. [67]
	SAT problem	NP-complete	✗	✓	Ref. [68]
Microfluid computing	K-shortest paths problem	NP-hard	✗	✗	Ref. [21]
	Maximum clique problem	NP-complete	✗	✓	Ref. [3]
Molecular computing	Subset sum problem	NP-complete	✓	✗	Ref. [8]
	SAT problem	NP-complete	✗	✓	Ref. [70]
	Knight's tour problem	NP-complete	✗	✓	Ref. [69]
	Hamiltonian path problem	NP-complete	✗	✓	Ref. [71]

waveguides inside the chip refer to the possible edges of the given graph, while the output waveguides correspond to all possible subgraphs. The input waveguides inside the chip are parallel, equally spaced input waveguides buried in the chip at the same depth of 170 μm, with a spacing of 127 μm. Each input waveguide successively connects to a single-layer waveguide network of different depth in the chip by bending along the three directions of the XYZ axis. To offset the effect of different bending losses between the input waveguides, we slightly adjust the input power of the different channels to ensure that each layer of the waveguide network receives the same intensity. The spacing between adjacent layers is 25 μm to guarantee no crosstalk between the different layers. In each single-layer waveguide network, the unit spacing of the waveguides in the Y-axis direction is 25 μm. For solving the maximum clique problem of different graphs, the waveguide network inside our chip is universal, and the difference lies in the photon injection conditions. The only limitation is the number of vertices, which requires that the vertices of the graph do not exceed the maximum case that our photonic chip can address. Since the solution of the few-vertex graph is contained in the solution space of the multivertex graph, our customized photonic chip is downward compatible in terms of the number of vertices.

The FPGA module of the photonic processor is homemade in our laboratory and its maximum clock rate is 100 MHz. The programming unit uses the continuous laser with a central wavelength of 780 nm and a linewidth of 2 nm. The photon beam emitted by the laser is transmitted through free space to a polarization beam splitter (PBS) with an extinction ratio of 1000:1. Then the P-polarized photons selected by PBS are coupled into a single-mode polarization-maintaining fiber patch cable through the collimation coupler. After that, the photons are injected into the input waveguide inside the three-dimensional photonic chip via a V-groove fiber array. The waveguide spacing of the V-groove is 127 μm, which matches the input waveguide spacing inside our chips. The V-groove array is attached to the side of the chip by end-face coupling and is cured by ultraviolet glue to ensure stability. Finally, the photons emitted from the end of the chip are converged into the CCD by two lenses with focal lengths of 150 and 200 mm, respectively.

APPENDIX C: PHOTONIC PROCESSOR OPERATION STEPS

Taking the given five-vertex graph shown in Fig. 8(a) as an example, we show the operation steps of the photonic

processor. Before conducting the experiments, we configure a series of lasers encoded in the programming unit. The lasers manifest a one-to-one correspondence with all possible edges on the graph to be solved, as shown in Figs. 8(a) and 8(b). Then the specific solving details, and the experimental operation process is as follows.

As shown in Figs. 8(a) and 8(b), the first thing is to enumerate all possible edges on the graph to be solved. Then those edges that exist on the given graph are recorded as 1 and the nonexistent edges recorded as 0. Thus, a set of digital signal codes is generated. Finally, the digital signals are fed into the FPGA. In this way, we have completed the input process of the graph information.

Next, the FPGA will generate multichannel high- and low-level signals to control the switching of the lasers based on the received digital signals. As shown in Fig. 8(b), 1 corresponds to a high level, which turns on the laser and 0 corresponds to a low level, which turns off the laser. Thus, we convert the edge information on the given graph into the injection status information of photons.

Subsequently, the optical signal from the programming unit is further coupled into the corresponding encoded input waveguide through the guidance of the V-groove single-mode fiber. The photons carrying the encoded information are then evolved in the three-dimensional photonic chip, i.e., to implement the problem solving process, as shown in Fig. 8(c). Then, the photons carrying the calculation results are emitted from the output waveguide of the chip and then recorded by the CCD camera of the readout unit.

Finally, based on the CCD recording of photon arrival information shown in Fig. 8(d), all cliques on the given graph can be identified and the solution of the maximum clique problem can be obtained.

APPENDIX D: SIGNAL-TO-NOISE RATIO ANALYSIS

The signal-to-noise ratio is a key metric for evaluating the performance of our photonic processors. In our scheme, the photons injected into the waveguide will, in principle, be emitted from the output ports in equal proportions. However, due to imperfections in chip fabrication and the influence of waveguide loss, the actual distribution of photon emission results may have some deviation from the theoretical value.

According to the definition of signal-to-noise ratio:

$$\mathrm{SNR} = 10\log_{10}\frac{\mathrm{Sig}}{N_{\mathrm{o\,i}}} \tag{D1}$$

$$= 10\log_{10}\frac{P_{\mathrm{i\,n}}}{N_{\mathrm{o\,i}}}\frac{\mathrm{Sig}}{P_{\mathrm{i\,n}}} \tag{D2}$$

$$= 10\log_{10}\frac{P_{\mathrm{i\,n}}}{N_{\mathrm{o\,i}}} - 10\log_{10}\frac{P_{\mathrm{i\,n}}}{\mathrm{Sig}}, \tag{D3}$$

where Sig is the signal power, $N_{\mathrm{o\,i}}$ is the ambient noise power, and $P_{\mathrm{i\,n}}$ is the input power.

Since the signal power is mainly affected by the splitting-ratio deviation and the waveguide loss, and the waveguide loss is an inherent property of the chip. Therefore, when evaluating the effect of imperfect fabrications on the experimental results, we mainly consider the influence of the nonideal beam-splitting ratio during the machining process on the signal power. Assuming that the beam-splitting network has a total of N waveguide output ports, the splitting ratio of the ith beam-splitting waveguide has a deviation δ_i from our expected value. According to our actual machining accuracy, the variation range of δ_i is about $(-0.05, 0.05)$. Moreover, we introduce a parameter σ to characterize the systematic error of the beam splitting ratio in the waveguide. If σ is positive, then the actual beam-splitting ratio (input waveguide intensity/coupled waveguide intensity) is greater than the ideal beam-splitting ratio, i.e., the intensity is biased towards the input waveguide, and vice versa. Using Sig_i to represent the output power of the ith waveguide, there are

$$\mathrm{Sig}_1 = \frac{P_{\mathrm{i\,n}}}{N}(1-\delta_1)(1+\sigma) \tag{D4}$$

$$\mathrm{Sig}_2 = \left(1-\mathrm{Sig}_1\right)\frac{1}{N-1}(1-\delta_2)(1+\sigma) \tag{D5}$$

$$\cdots$$

$$\mathrm{Sig}_i = \left(1-\mathrm{Sig}_1-\mathrm{Sig}_2-\cdots-\mathrm{Sig}_{i-1}\right) \times \frac{1}{N+1-i}(1-\delta_i)(1+\sigma) \tag{D6}$$

$$\cdots$$

$$\mathrm{Sig}_{N-1} = \left(1-\mathrm{Sig}_1-\mathrm{Sig}_2-\cdots-\mathrm{Sig}_{N-2}\right) \times \frac{1}{2}(1-\delta_{N-1})(1+\sigma) \tag{D7}$$

$$\mathrm{Sig}_N = 1-\sum_{i=1}^{N-1}\mathrm{Sig}_i \tag{D8}$$

$$\delta_i = \mathrm{Random}(-0.05, 0.05) \tag{D9}$$

$$\sigma = \mathrm{Constant}. \tag{D10}$$

As the systematic errors are nonrandom, deterministic, and directional, it is worthwhile to consider the sign of σ when simulating the systematic errors. As shown in Figs. 9(a)–9(f), we simulate the power distribution of each output waveguide when N is taken as 500 and 1000, $\sigma = 0$, $\sigma = 0.03$, and $\sigma = -0.03$, respectively. The numerical simulation results demonstrate that when $\sigma = 0$, the fluctuation of Sig_i is only within a small range around the expected value, and Δ decreases as N increases. Therefore, our scheme is robust against the influence of random

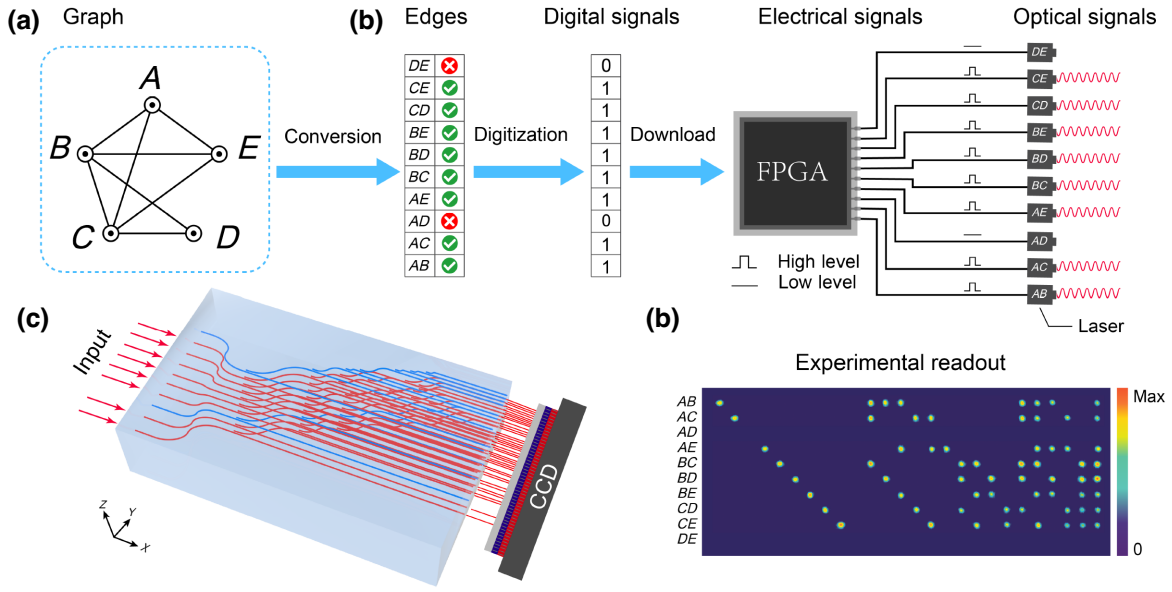

FIG. 8. Schematic diagram of the actual operating process and workflow of the photonic processor. (a) A randomly given five-vertex graph to be solved. (b) Schematic diagram of the input signal transmission and conversion process. Comparing the edges present on the given graph with all possible edges of the five-vertex graph, a set of digital signals consisting of 0, 1 can be obtained, where 0 indicates the absence of the corresponding edge and vice versa. The digital signals are then fed into the FPGA, and the FPGA generates corresponding high and low voltages according to the digital signals and applies them to a series of tagged lasers. Different lasers generate two states of ON or OFF based on the received high and low voltages. This completes the process of signal input and conversion. (c) Schematic diagram of the process where photons carrying information enter a photonic processor to perform computational operations. (d) Experimental record of the distribution of optical spots captured by a CCD camera.

errors. Positive σ values bias the output intensity of the upstream waveguide to be larger and that of the downstream waveguide to be smaller, while negative σ values have the opposite effect. Both positive and negative σ values have a much greater effect on the downstream waveguide than on the upstream waveguide. This is because, with the same σ value, the value of the beam-splitting ratio for the downstream waveguide is much larger than that of the upstream waveguide, resulting in a higher absolute error. In summary, the deviation of the output intensity of the waveguides from the ideal value is the result of the combined effect of the random and systematic errors in the beam-splitting ratio.

In our scheme, the accuracy of the calculation depends only on whether the weakest beam signal can be read. Therefore, when considering the overall signal-to-noise ratio of the system, we need only to consider the minimum power output of the waveguide, that is $\mathrm{Sig}_{\min} = \frac{1}{N}P_{\mathrm{in}}(1-\Delta_{\max})$, where $\Delta_{\max}$ represents the proportion of the weakest light intensity deviation from the expected value in the output waveguide.

Based on the above analysis, we simulate the trend of $\Delta_{\max}$ with the increase of the number of waveguides under the conditions of $\delta_i = \mathrm{R\,a\,n\,d\,o\,m}(-0.05, 0.05)$ and σ being 0.1, 0.05, 0.03, 0, -0.03, -0.05, and -0.1, as shown in Figs. 10(a) and 10(b). According to the simulation results, we can see that the sign of σ has a huge impact on $\Delta_{\max}$. A positive value of σ will cause $\Delta_{\max}$ to increase, which seriously reduces the signal-to-noise ratio of our processor. On the other hand, a negative value of σ stabilizes the value of $\Delta_{\max}$. In view of the fact that systematic errors can be corrected and eliminated through system calibration, compensation, etc. This also inspires us that when designing the splitting ratio of the waveguide, the influence of systematic errors should be fully considered. The deviation of the splitting ratio caused by systematic errors can be compensated by slightly adjusting the coupling length of the waveguide to offset its effect.

Then we analyze the effect of the actual chip loss on the signal-to-noise ratio. We introduce a parameter τ that characterizes the waveguide loss. The loss per unit length of the waveguide can be approximately considered to be the same. The waveguide length increases linearly with N. According to the above analysis results, we have

$$\mathrm{S\,N\,R} = 10\log_{10}\frac{P_{\mathrm{i\,n}}}{N_{\mathrm{o\,i}}} - 10\log_{10}\frac{P_{\mathrm{i\,n}}}{S_{ig_{\min}}} - \tau N \qquad \text{(D11)}$$

$$= 10\log_{10}\frac{P_{\mathrm{i\,n}}}{N_{\mathrm{o\,i}}} - 10\log_{10}\frac{N}{1-\Delta} - \tau N. \qquad \text{(D12)}$$

The above formula can be simplified to the following form:

$$\mathrm{S\,N\,R}(N) = S - 10\log_{10}N - \mu_1 N - \mu_2, \qquad \text{(D13)}$$

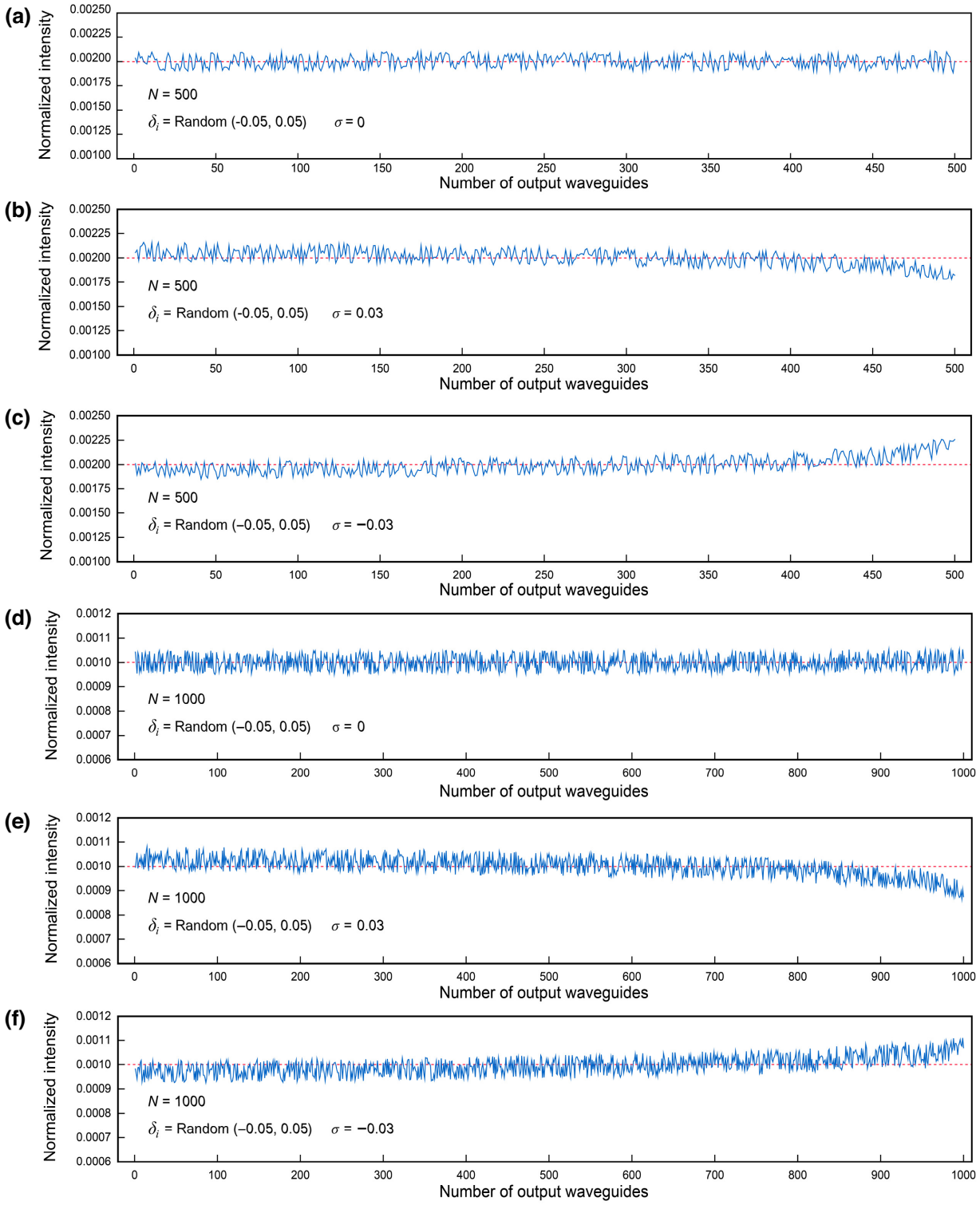

FIG. 9. Influence of nonideal splitting ratio on waveguide outputs. Under the premise that the random error δ_i varies in the range $(-0.05, 0.05)$, we simulate the distribution of the normalized power of the output with the number of output waveguides N and the systematic error σ, respectively. (a)–(c) Normalized power distribution of the waveguide output when the number of waveguides $N = 500$ and the systematic errors σ are 0, 0.03, -0.03, respectively. (d)–(f) Normalized power distribution of the waveguide output when the number of waveguides $N = 1000$ and the systematic errors σ are 0, 0.03, -0.03, respectively.

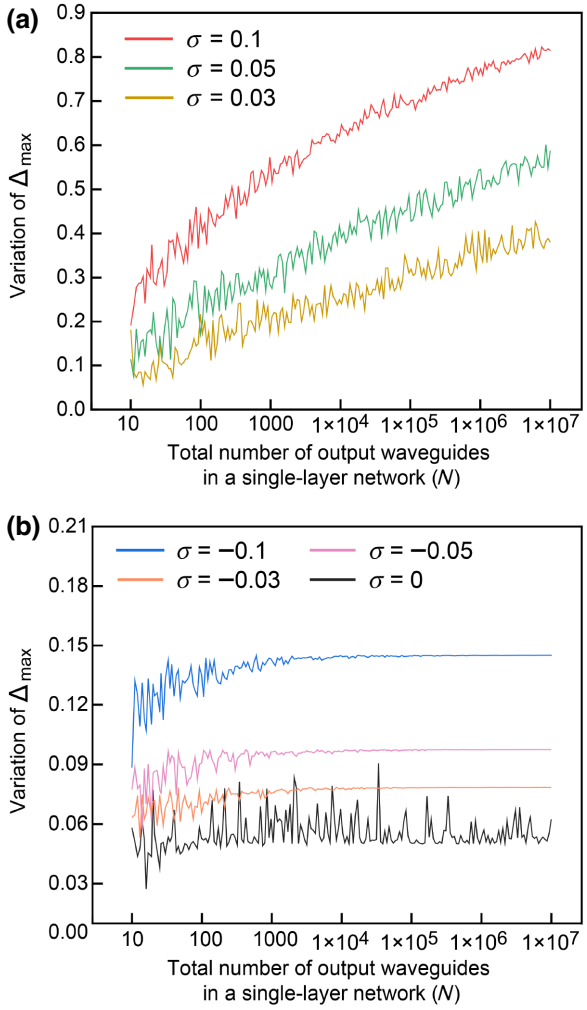

FIG. 10. The changing trend of the influence of different σ on Δ_{max} with the increase of the number of waveguides. (a) The effect of a positive σ value on Δ_{max} becomes larger and larger as the number of waveguides increases. (b) The effect of negative σ values on Δ_{max} tends to stabilize as the number of waveguides increases. When $\sigma = 0$, Δ_{max} fluctuates randomly in a small range. In general, the larger the absolute value of σ, the larger the value of Δ_{max}.

where $S = 10\log_{10} P_{in}/N_{0i}$, $\mu_1 = \tau$, and $\mu_2 = 10\log_{10} 1/(1-\Delta)$. μ_1 and μ_2 are experimental parameters related to our photonic processor, which characterize the effects of chip loss and imperfect fabrications on the signal-to-noise ratio, respectively. Here, according to the actual scenarios in our experiments, μ_1 and μ_2 are evaluated as $2.5 * 10^{-4}$ and $2.2 * 10^{-1}$, respectively.

We further evaluate the variation of the SNR with N. The first term of Eq. (D13) is a constant independent of N, which determines the upper limit of the signal-to-noise ratio. The latter two items decrease as N increases. Take a silicon detector with a dark count of 100 Hz and a detection efficiency of 75% as an example. We analyze the change of signal-to-noise ratio with the increase of N under

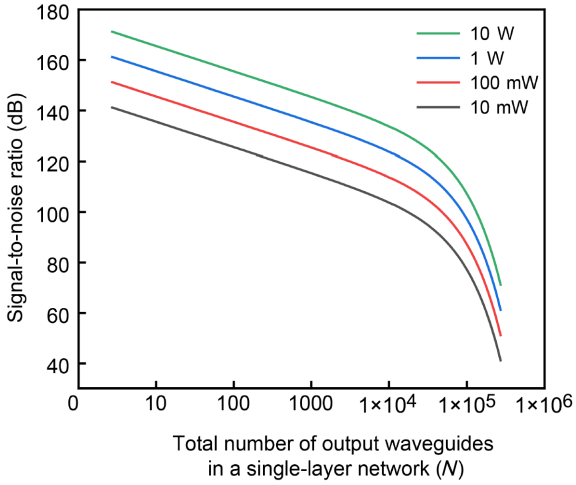

FIG. 11. Signal-to-noise ratio. Trend of the signal-to-noise ratio of the photonic chip with increasing number of waveguides at different input powers.

different input power conditions, as depicted in Fig. 11. Our results show that the higher the input power, the better the SNR, and the SNR decreases with the increase of N, and the decreasing trend is independent of the input power.

APPENDIX E: EXPERIMENTAL DEMONSTRATIONS OF VARIOUS FIVE- AND SIX-VERTEX GRAPHS

Except for the two cases listed in the main text. We also randomly generate different five- and six-vertex graphs and solve them with our photonic processor. Benefiting from the programmability of our processor, we can solve various graphs only by changing different photon injection conditions. Taking our experimentally fabricated chip for solving the six-vertex graph as an example, the chip has a total of $\binom{6}{2} = 15$ independently controlled input terminals, and each input terminal has two different input conditions, so there are a total of $2^{15} = 32\,768$ different input conditions. Except for the special case of an empty set, this photonic chip can solve the maximum clique problem for up to 32 767 different graphs. As exhibited in Figs. 12(a)–12(h) and 13(a)–13(f), we present some experimental results of various five- and six-vertex graphs, respectively. Our results suggest that our photonic processor has scalability, stability, and flexible programmability. At the same time, it is confirmed that it is possible to realize the programming scheme by controlling the injection condition of photons.

APPENDIX F: TIME COMPLEXITY OF THE MCP AND EVALUATION OF COMPUTING TIME

In general, for an n-vertex graph, there are 2^n different combinations of vertices that can form subgraphs.

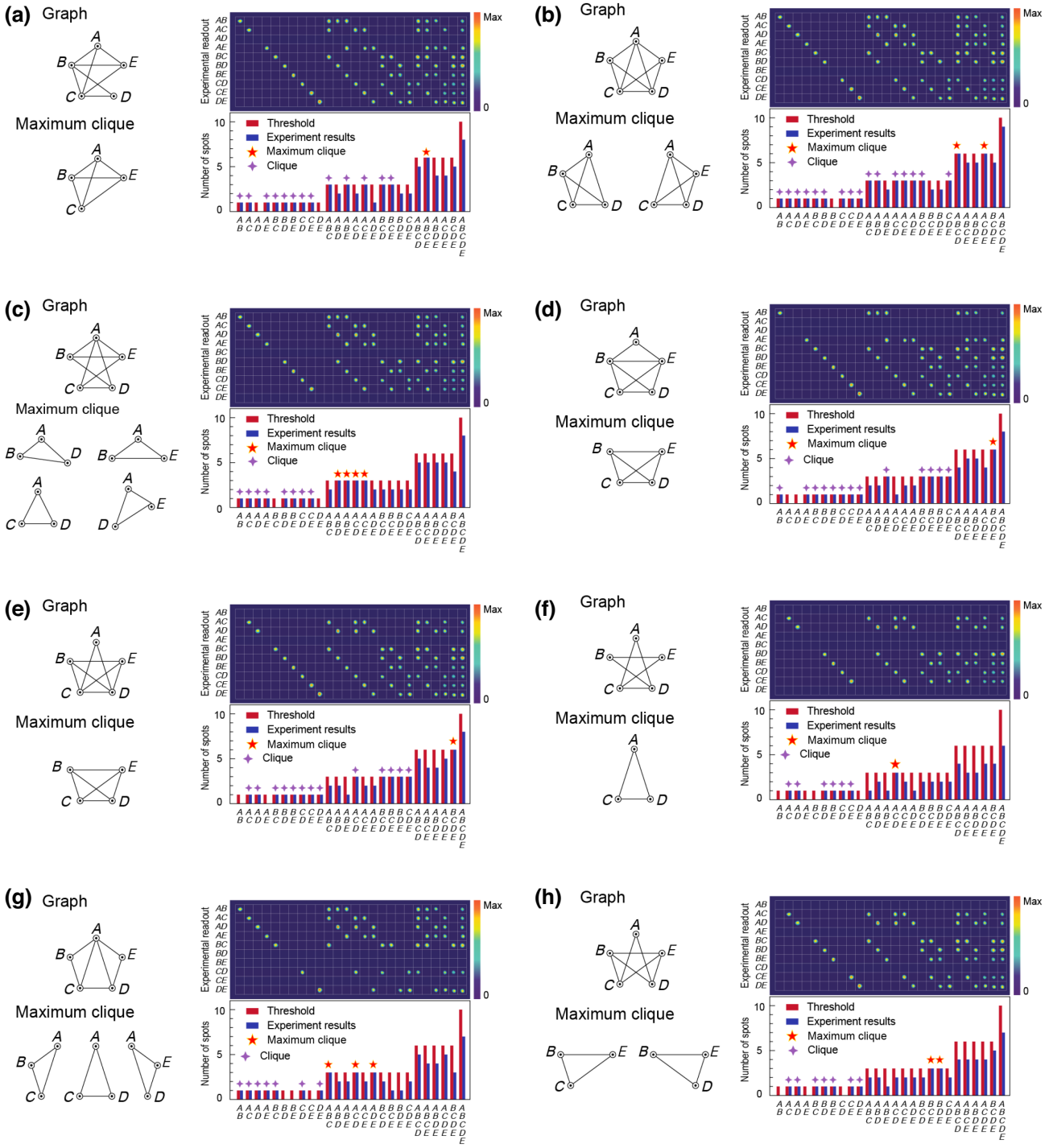

FIG. 12. Demonstration of experimental results for various five-vertex graphs. (a)–(h) are the experimental results of our programmable photonic processor for solving the maximum clique problem of random five-vertex graphs, respectively.

Naturally exclude $(n+1)$ scenarios that are trivial [3]: one empty set and the n subgraphs that contain only one vertex, that is, there are $(2^n - n - 1)$ subgraphs to be checked. In each check, for a k-vertex subgraph, $\binom{k}{2} = k(k-1)/2$ judgments are required to determine whether all vertices are connected in that subgraph. Therefore, using the enumeration algorithm to solve the MCP problem requires a total of $\sum_{k=1}^{n}\left[\binom{n}{k}\frac{k(k-1)}{2}\right]$ operations, and the time complexity of the algorithm is $\Omega\left(k^2\binom{n}{k}\right)$ [73].

With the progress of algorithm research, some optimization algorithms for solving the MCP have been proposed. The currently optimal algorithm is a depth-first search method based on the Bron-Kerbosch algorithm [74]

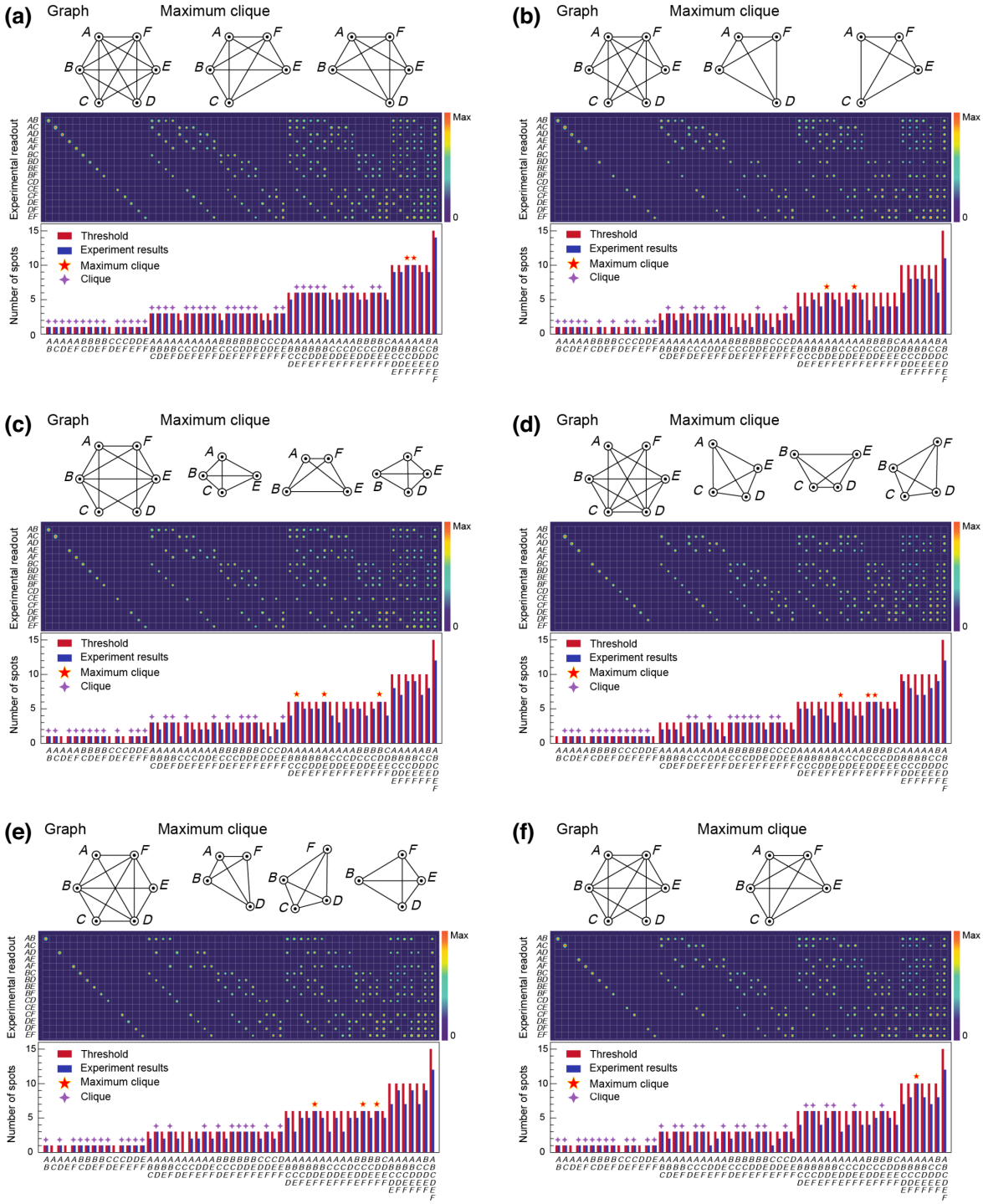

FIG. 13. Demonstration of experimental results for various six-vertex graphs. (a)–(f) are the experimental results of our programmable photonic processor for solving the maximum clique problem of random six-vertex graphs, respectively.

proposed by Tomita *et al*. It can enumerate all cliques on a given graph and reduce the time complexity to $O\left(3^{n/3}\right)$. However, this algorithm still cannot get rid of the exponentially increasing time consumption. Moreover, during the calculation, only the subgraphs satisfying the clique condition are kept, and other subgraphs are discarded. As a result, the optimization algorithm cannot obtain all the subgraph data, which is key information in graph analysis [75,76].

The computation time of the photonic processor is the sum of the time consumption of the programming unit, the processing unit and the readout unit. The time

consumption of the programming unit and the readout unit is not sensitive to the increase of the problem size, and their computation time can be evaluated based on hardware performance. The computing time of the processing unit mainly depends on the propagation speed of the photons and the longest path length in the chip [9]. Note that there is no intersection between the waveguide networks of different depths in our chip, so we can divide the waveguide network into multiple square subnetworks with the maximum vertical depth of the waveguide network as the side length, and use multiple ports to simultaneously inject photons to meet the challenges of increasing problem size. According to our waveguide design rules combined with the network segmentation co-working scheme, the longest path lengths taken by photons can be estimated. The propagation speed of photons in the chip is about 2×10^8 m/s, which is estimated by the change of refractive index of the chip induced by the femtosecond laser.

The computation time of electronic computers is evaluated according to the calculation amount of the actual computing task and the number of theoretical floating-point operations per second (FLOPS) [34–36]. We assume that electronic computers operate at peak computing performance, and ignore the possible degradation of computing speed caused by the long-term operation of the electronic computers in real scenarios.

APPENDIX G: PHARMACOPHORE MODEL

The concept of pharmacophore [77] was first proposed by Paul Ehrlich in 1909, and has since then been developed continuously. Pharmacophore refers to a group of structural features in a molecule that can recognize receptors and form molecular biological activity. It is an abstract description of molecular features. Common types of pharmacophores include hydrogen-bond donors, hydrogen-bond acceptors, positively charged centers, negatively charged centers, hydrophobic centers, aromatic ring centers, etc. [45,46,48].

Hydrogen-bond donors [78–80] mainly consist of hydrogen atoms and the oxygen and nitrogen atoms attached to them. The common groups that can form hydrogen bonds are as follows: (i) nonacidic hydroxyl groups; (ii) amino groups; (iii) the secondary amino group, excluding the secondary amino group in trifluoromethanesulfonamide and tetrazole.

Hydrogen-bond acceptors [78–80] mainly refer to atoms with lone pairs of electrons. The common forms of hydrogen-bond acceptors are as follows: (i) oxygen atoms with sp or sp^2 hybridization; (ii) sulfur atom attached to carbon atom by double bond; (iii) nitrogen atoms with a double or triple bond to a carbon atom.

Positive charge centers [79,80] mainly contain the following: (i) positively charged atoms; (ii) nitrogen atoms in primary, secondary, or tertiary aliphatic amines; (iii) the iminonitrogen atom in the nitrogen-nitrogen disubstituted amidine group or iminonitrogen atoms in tetranitrogen-substituted guanidinium groups; (iv) a nitrogen atom center in an amidine group containing at least one unsubstituted hydrogen atom or a nitrogen atom center in a guanidine group containing at least one unsubstituted hydrogen atom.

Negative charge centers [79,80] mainly contain the following: (i) negatively charged atoms; (ii) nitrogen atom in trifluoromethanesulfonamide; (iii) atomic centers of hydroxyl and oxo oxygen in carboxylic acid, sulfinic acid, or phosphoric acid; (v) atomic centers of hydroxyl and oxo oxygen in phosphoric acid diesters and phosphate esters; (vi) the atomic centers of hydroxyl and two oxo oxygen in sulfuric acid and sulfonic acid; (vii) atomic centers of oxo oxygen and two hydroxyl oxygen in phosphate monoester and phosphate; (viii) amino nitrogen atom in tetrazole.

Hydrophobic groups [79,80] are generally composed of nonpolar atoms, and common hydrophobic fragments are methyl, ethyl, etc.

The aromatic ring centers [79,80] mainly include five- or six-membered aromatic rings, such as thiophene and benzene rings.

APPENDIX H: MATCHING CHEMISTRY AND GEOMETRICAL SHAPE IN MOLECULAR DOCKING

Molecular docking originates from the "lock-and-key" model [81,82], which is rough but still instructive. The real molecular docking process is often more complicated. Firstly, molecules are flexible to a certain extent, not completely rigid. Secondly, the molecular docking process not only needs to consider the geometric complementarity of space, but also needs to consider the chemical matching between the pharmacophore. Molecular docking is the process of mutual recognition between ligand and receptor molecules through chemical matching and spatial geometric complementarity. Therefore, in the process of molecular docking, the following matching principles are mainly followed between ligands and receptors:

(1) Hydrogen-bonding interactions [82].

Due to the large dipole moment of the groups formed by the negatively charged atoms and hydrogen, the distribution of the bonding electron cloud is biased towards the more negatively charged atoms. As a result, there are fewer electrons distributed around the hydrogen nucleus, and the positively charged hydrogen nucleus (proton) is exposed outside. When this positively charged hydrogen nucleus encounters another atom with strong electronegativity, electrostatic attraction occurs, which is called hydrogen bonding. Hydrogen bonds play a key role in maintaining the structure of proteins.

(2) Electrostatic interactions [82].

An electrostatic interaction between positive and negative charges in a ligand and receptor. The attractive force F is proportional to the product of the charge $(q_1 * q_2)$ and inversely proportional to the square of the distance between the charged particles (r^2), and decreases with the increase of the dielectric constant of the surrounding medium in the solution.

(3) Hydrophobic interactions [82].

The hydrophobic center is generally composed of nonpolar atoms. In the process of mutual recognition between ligands and receptors, the hydrophobic groups on ligands and receptors always tend to bury hydrophobic residues in the interior of the molecule to form a tight hydrophobic core. The essence of hydrophobic interaction is not the attraction between hydrophobic groups, but the interaction between hydrophobic groups or hydrophobic side chains due to the forced approach of hydrophobic groups.

(4) $\pi - \pi$ interactions [82].

$\pi - \pi$ interaction, also known as $\pi - \pi$ stacking, refers to an interaction that often occurs between aromatic rings, usually between two molecules that are relatively electron rich and electron deficient, which are useful noncovalent interactions.

(5) Complementary matching of geometric shapes [46].

Geometric complementarity requires tight binding of ligands and receptors with a large contact area. Therefore, in the process of molecular docking, it is necessary to find as many paired pharmacophore pairs on the ligand and receptor as possible. Considering that there is a certain degree of flexibility between actual molecules, we introduce a parameter ε to characterize the flexibility of the molecules. When the difference between the Euclidean distance between the two pharmacophores on the ligand and the Euclidean distance between the two pharmacophores on the receptor is less than ε, we consider that the two groups of pharmacophore pairs satisfy the spatial matching relation.

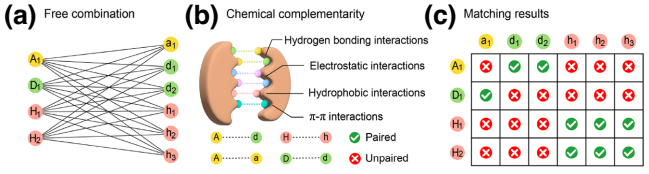

FIG. 14. Chemical matching between pharmacophores. (a) All possible combinations of pharmacophore on ligand and receptor under free binding. (b) Principles of chemical matching of pharmacophore pairs on receptor and ligand. (c) The matching results obtained according to the chemical matching rules shown in (b).

Euclidean distance difference between two pairs of pharmacophores (Å)

	(A_1,d_1)	(A_1,d_2)	(D_1,a_1)	(H_1,h_1)	(H_1,h_2)	(H_1,h_3)	(H_2,h_1)	(H_2,h_2)	(H_2,h_3)
(A_1,d_1)		●	1.268	9.701	12.376	11.707	4.944	7.619	6.950
(A_1,d_2)	●		4.396	2.777	5.524	12.768	1.980	0.767	8.011
(D_1,a_1)	✓	✓		1.691	4.195	10.172	0.800	1.704	7.681
(H_1,h_1)	✗	✓	✓		●	●	●	2.798	4.558
(H_1,h_2)	✗	✗	✗	●		●	2.798	●	1.577
(H_1,h_3)	✗	✗	✗	●	●		4.558	1.577	●
(H_2,h_1)	✗	✓	✓	●	✓	✗		●	●
(H_2,h_2)	✗	✓	✓	✓	●	✓	●		●
(H_2,h_3)	✗	✗	✗	✗	✓	●	●	●	

✓ Match ✗ Mismatch ● Contradictory

FIG. 15. Spatial geometry matching between pharmacophore pairs. Euclidean distance difference data between any two pairs of pharmacophores, and spatial matching results. There are 12 sets of pharmacophore pairs satisfying the spatial matching condition.

APPENDIX I: DOCKING DETAILS OF THE RECEPTOR (3MZ3) AND THE LIGAND (B3N)

As shown in Fig. 14(a), there are 24 possible contact types for receptor 3MZ3 and the ligand B3N under the free combination condition. However, not every combination formed by contact is reasonable. According to the chemical matching rules [45,48] between the pharmacophores, we exclude the pairs of pharmacophores that do not satisfy the chemical match and obtain nine pairs of paired pharmacophore pairs, as shown in Fig. 14(c).

To investigate the possible spatial match between these nine pharmacophore pairs, we enumerated all combinations, as shown in Fig. 15. But a pharmacophore on a ligand or receptor can only participate in one pharmacophore pair, so we use gray to distinguish the contradictory cases. After calculation, we obtained the Euclidean distance difference $|L_m - L_n|$ data between different pairs of pharmacophores [see the upper right corner of Fig. 15]. Based on the spatial matching flexibility threshold, we obtain 12 sets of pharmacophore pairs that satisfy the spatial compatibility [see the lower left corner of Fig. 15].

[1] R. Lambiotte, P. L. Krapivsky, U. Bhat, and S. Redner, Structural transitions in densifying networks, Phys. Rev. Lett. **117**, 218301 (2016).

[2] Z. Lu, J. Wahlström, and A. Nehorai, Community detection in complex networks via clique conductance, Sci. Rep. **8**, 5982 (2018).

[3] D. T. Chiu, E. Pezzoli, H. Wu, A. D. Stroock, and G. M. Whitesides, Using three-dimensional microfluidic networks for solving computationally hard problems, Proc. Natl. Acad. Sci. **98**, 2961 (2001).

[4] A. E. Moylett, N. Linden, and A. Montanaro, Quantum speedup of the traveling-salesman problem for bounded-degree graphs, Phys. Rev. A **95**, 032323 (2017).

[5] R. H. Warren, Solving the traveling salesman problem on a quantum annealer, SN Appl. Sci. **2**, 75 (2020).

[6] S. Marsh and J. B. Wang, A quantum walk-assisted approximate algorithm for bounded NP optimisation problems, Quantum Inf. Process. **18**, 1 (2019).

[7] E. Pelofske, G. Hahn, and H. Djidjev, in *Proceedings of the 16th ACM International Conference on Computing Frontiers* (Association for Computing Machinery, New York, 2019), p. 76.

[8] D. V. Nicolau Jr, M. Lard, T. Korten, F. C. Van Delft, M. Persson, E. Bengtsson, A. Månsson, S. Diez, H. Linke, and D. V. Nicolau, Parallel computation with molecular-motor-propelled agents in nanofabricated networks, Proc. Natl. Acad. Sci. **113**, 2591 (2016).

[9] X.-Y. Xu, X.-L. Huang, Z.-M. Li, J. Gao, Z.-Q. Jiao, Y. Wang, R.-J. Ren, H. P. Zhang, and X.-M. Jin, A scalable photonic computer solving the subset sum problem, Sci. Adv. **6**, eaay5853 (2020).

[10] K. Wu, J. García de Abajo, C. Soci, P. Ping Shum, and N. I. Zheludev, An optical fiber network oracle for NP-complete problems, Light: Sci. Appl. **3**, e147 (2014).

[11] F. L. Traversa, C. Ramella, F. Bonani, and M. Di Ventra, Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states, Sci. Adv. **1**, e1500031 (2015).

[12] J. Von Neumann and R. Kurzweil, *The Computer and the Brain* (Yale University Press, Connecticut, 2012).

[13] B. Sun, T. Guo, G. Zhou, S. Ranjan, Y. Jiao, L. Wei, Y. N. Zhou, and Y. A. Wu, Synaptic devices based neuromorphic computing applications in artificial intelligence, Mater. Today Phys. **18**, 100393 (2021).

[14] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[15] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar *et al.*, Quantum optimization of maximum independent set using Rydberg atom arrays, Science **376**, 1209 (2020).

[16] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins *et al.*, Quantum computational advantage with a programmable photonic processor, Nature **606**, 75 (2022).

[17] J. Gao, X.-W. Wang, W.-H. Zhou, Z.-Q. Jiao, R.-J. Ren, Y.-X. Fu, L.-F. Qiao, X.-Y. Xu, C.-N. Zhang, X.-L. Pang, Hang Li Yao Wang, and Xian-Min Jin, Quantum advantage with membosonsampling, Chip **1**, 100007 (2022).

[18] Q. Liu, L. Wang, A. G. Frutos, A. E. Condon, R. M. Corn, and L. M. Smith, DNA computing on surfaces, Nature **403**, 175 (2000).

[19] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. Adleman, Solution of a 20-variable 3-SAT problem on a DNA computer, Science **296**, 499 (2002).

[20] J.-J. Shu, Q.-W. Wang, and K.-Y. Yong, DNA-based computing of strategic assignment problems, Phys. Rev. Lett. **106**, 188702 (2011).

[21] D. R. Reyes, M. M. Ghanem, G. M. Whitesides, and A. Manz, Glow discharge in microfluidic chips for visible analog computing, Lab Chip **2**, 113 (2002).

[22] R. Prevedel, P. Walther, F. Tiefenbacher, P. Böhi, R. Kaltenbaek, T. Jennewein, and A. Zeilinger, High-speed linear optics quantum computing using active feed-forward, Nature **445**, 65 (2007).

[23] Z. Ying, C. Feng, Z. Zhao, S. Dhar, H. Dalir, J. Gu, Y. Cheng, R. Soref, D. Z. Pan, and R. T. Chen, Electronic-photonic arithmetic logic unit for high-speed computing, Nat. Commun. **11**, 2154 (2020).

[24] X.-B. Xu, W.-T. Wang, L.-Y. Sun, and C.-L. Zou, Hybrid superconducting-photonic-phononic chip for quantum information processing, Chip **1**, 100016 (2022).

[25] L.-T. Feng, M. Zhang, Z.-Y. Zhou, M. Li, X. Xiong, L. Yu, B.-S. Shi, G.-P. Guo, D.-X. Dai, X.-F. Ren *et al.*, On-chip coherent conversion of photonic quantum entanglement between different degrees of freedom, Nat. Commun. **7**, 11985 (2016).

[26] J. Wang, F. Sciarrino, A. Laing, and M. G. Thompson, Integrated photonic quantum technologies, Nat. Photonics **14**, 273 (2020).

[27] J. Lebreuilly, A. Biella, F. Storme, D. Rossini, R. Fazio, C. Ciuti, and I. Carusotto, Stabilizing strongly correlated photon fluids with non-Markovian reservoirs, Phys. Rev. A **96**, 033828 (2017).

[28] J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh *et al.*, Universal linear optics, Science **349**, 711 (2015).

[29] X. Qiang, X. Zhou, J. Wang, C. M. Wilkes, T. Loke, S. O'Gara, L. Kling, G. D. Marshall, R. Santagati, T. C. Ralph *et al.*, Large-scale silicon quantum photonics implementing arbitrary two-qubit processing, Nat. Photonics **12**, 534 (2018).

[30] Y. Hu, M. Yu, D. Zhu, N. Sinclair, A. Shams-Ansari, L. Shao, J. Holzgrafe, E. Puma, M. Zhang, and M. Lončar, On-chip electro-optic frequency shifters and beam splitters, Nature **599**, 587 (2021).

[31] A. Crespi, R. Ramponi, R. Osellame, L. Sansoni, I. Bongioanni, F. Sciarrino, G. Vallone, and P. Mataloni, Integrated photonic quantum gates for polarization qubits, Nat. Commun. **2**, 566 (2011).

[32] K. Okamoto, *Fundamentals of Optical Waveguides* (Elsevier, Amsterdam, 2021).

[33] D. Guzmán-Silva, G. Cáceres-Aravena, and R. A. Vicencio, Experimental observation of interorbital coupling, Phys. Rev. Lett. **127**, 066601 (2021).

[34] Intel Xeon Platinum 8362 Processor, www.intel.com/.

[35] Nivida GeForce RTX 3090Ti, www.nvidia.com/en-us/.

[36] Supercomputer: Fugaku, www.top500.org/.

[37] S. Busygin, A new trust region technique for the maximum weight clique problem, Discrete Appl. Math. **154**, 2080 (2006).

[38] Q. Wu, J.-K. Hao, and F. Glover, Multi-neighborhood Tabu search for the maximum weight clique problem, Ann. Oper. Res. **196**, 611 (2012).

[39] J. Pattillo, A. Veremyev, S. Butenko, and V. Boginski, On the maximum quasi-clique problem, Discrete Appl. Math. **161**, 244 (2013).

[40] B. Q. Pinto, C. C. Ribeiro, I. Rosseti, and A. Plastino, A biased randomkey genetic algorithm for the maximum quasi-clique problem, Eur. J. Oper. Res. **271**, 849 (2018).

[41] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (Association for Computing Machinery, New York, 2013), p. 104.

[42] M. A. U. Nasir, A. Gionis, G. D. F. Morales, and S. Girdzijauskas, in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Association for Computing Machinery, New York, 2017), p. 1817.

[43] L. G. Ferreira, R. N. Dos Santos, G. Oliva, and A. D. Andricopulo, Molecular docking and structure-based drug design strategies, Molecules **20**, 13384 (2015).

[44] S. Agarwal and R. Mehrotra, An overview of molecular docking, JSM Chem **4**, 1024 (2016).

[45] A. R. Leach, V. J. Gillet, R. A. Lewis, and R. Taylor, Three-dimensional pharmacophore methods in drug discovery, J. Med. Chem. **53**, 539 (2010).

[46] L. Banchi, M. Fingerhuth, T. Babej, C. Ing, and J. M. Arrazola, Molecular docking with Gaussian Boson sampling, Sci. Adv. **6**, eaax1950 (2020).

[47] RCSB protein data bank, www.rcsb.org/.

[48] G. Wolber and T. Langer, LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters, J. Chem. Inf. Model. **45**, 160 (2005).

[49] S. Sarfaraz, I. Muneer, and H. Liu, Combining fragment docking with graph theory to improve ligand docking for homology model structures, J. Comput. Aided Mol. Des. **34**, 1237 (2020).

[50] J. A. Ruiz-Vanoye, J. Pérez-Ortega, R. A. Pazos R., O. Díaz-Parra, J. Frausto-Solís, H. J. Fraire Huacuja, L. Cruz-Reyes, and J. A. Martínez F., Survey of polynomial transformations between NP-complete problems, J. Comput. Appl. Math. **235**, 4851 (2011).

[51] M. M. Waldrop, The chips are down for Moore's law, Nat. News **530**, 144 (2016).

[52] K. Flamm, Has Moore's law been repealed? An economist's perspective, Comput. Sci. Eng. **19**, 29 (2017).

[53] D. Sarkar, X. Xie, W. Liu, W. Cao, J. Kang, Y. Gong, S. Kraemer, P. M. Ajayan, and K. Banerjee, A subthermionic tunnel field-effect transistor with an atomically thin channel, Nature **526**, 91 (2015).

[54] J. Shalf, The future of computing beyond Moore's law, Philos. Trans. R. Soc. A **378**, 20190061 (2020).

[55] R. S. Williams, What's next? [The end of Moore's law], Comput. Sci. Eng. **19**, 7 (2017).

[56] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[57] J. A. Filar, M. Haythorpe, and R. Taylor, Linearly-growing reductions of Karp's 21 NP-complete problems, arXiv preprint arXiv:1902.10349 (2019).

[58] D. S. Hochba, Approximation algorithms for NP-hard problems, ACM Sigact News **28**, 40 (1997).

[59] T. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemey *et al.*, Multi-qubit entanglement and algorithms on a neutral-atom quantum computer, Nature **604**, 457 (2022).

[60] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nat. Phys. **17**, 332 (2021).

[61] X. Qiang, Y. Wang, S. Xue, R. Ge, L. Chen, Y. Liu, A. Huang, X. Fu, P. Xu, T. Yi *et al.*, Implementing graph-theoretic quantum algorithms on a silicon photonic quantum walk processor, Sci. Adv. **7**, eabb8375 (2021).

[62] H. Wang, W. Li, X. Jiang, Y.-M. He, Y.-H. Li, X. Ding, M.-C. Chen, J. Qin, C.-Z. Peng, C. Schneider *et al.*, Toward scalable boson sampling with photon loss, Phys. Rev. Lett. **120**, 230502 (2018).

[63] H. Wang, Y. He, Y.-H. Li, Z.-E. Su, B. Li, H.-L. Huang, X. Ding, M.-C. Chen, C. Liu, J. Qin *et al.*, High-efficiency multiphoton boson sampling, Nat. Photonics **11**, 361 (2017).

[64] M. Tillmann, B. Dakić, R. Heilmann, S. Nolte, A. Szameit, and P. Walther, Experimental boson sampling, Nat. Photonics **7**, 540 (2013).

[65] J. B. Spring, B. J. Metcalf, P. C. Humphreys, W. S. Kolthammer, X.-M. Jin, M. Barbieri, A. Datta, N. Thomas-Peter, N. K. Langford, D. Kundys *et al.*, Boson sampling on a photonic chip, Science **339**, 798 (2013).

[66] L. Wang, J. G. Hall, M. Lu, Q. Liu, and L. M. Smith, A DNA computing readout operation based on structure-specific cleavage, Nat. Biotechnol. **19**, 1053 (2001).

[67] Q. Ouyang, P. D. Kaplan, S. Liu, and A. Libchaber, DNA solution of the maximal clique problem, Science **278**, 446 (1997).

[68] R. J. Lipton, DNA solution of hard computational problems, Science **268**, 542 (1995).

[69] D. Faulhammer, A. R. Cukras, R. J. Lipton, and L. F. Landweber, Molecular computation: RNA solutions to chess problems, Proc. Natl. Acad. Sci. **97**, 1385 (2000).

[70] K. Sakamoto, H. Gouzu, K. Komiya, D. Kiga, S. Yokoyama, T. Yokomori, and M. Hagiya, Molecular computation by DNA hairpin formation, Science **288**, 1223 (2000).

[71] L. M. Adleman, Molecular computation of solutions to combinatorial problems, Science **266**, 1021 (1994).

[72] H. Tang, X.-F. Lin, Z. Feng, J.-Y. Chen, J. Gao, K. Sun, C.-Y. Wang, P.-C. Lai, X.-Y. Xu, Y. Wan *et al.*, Experimental two-dimensional quantum walk on a photonic chip, Sci. Adv. **4**, eaat3174 (2018).

[73] A. Mani, G. Cattaneo, and I. Düntsch, *Algebraic Methods in General Rough Sets*, Springer International Publishing (Springer, New York, 2018).

[74] E. Tomita, A. Tanaka, and H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments, Theor. Comput. Sci. **363**, 28 (2006).

[75] V. Bonnici, R. Giugno, A. Pulvirenti, D. Shasha, and A. Ferro, A subgraph isomorphism algorithm and its application to biochemical data, BMC Bioinformatics **14**, 1 (2013).

[76] Y. Sun, G. Li, J. Du, B. Ning, and H. Chen, A subgraph matching algorithm based on subgraph index for knowledge graph, Front. Comput. Sci. **16**, 1 (2022).

[77] O. Guner, History and evolution of the pharmacophore concept in computer-aided drug design, Curr. Top. Med. Chem. **2**, 1321 (2002).

[78] G. R. Desiraju and T. Steiner, *The Weak Hydrogen Bond: In Structural Chemistry and Biology* (International Union of Crystal, Melbourne, 2001).

[79] R. Mannhold, H. Kubinyi, and G. Folkers, *Pharmacophores and Pharmacophore Searches* (John Wiley & Sons, New Jersey, 2006).

[80] K. Stromgaard, P. Krogsgaard-Larsen, and U. Madsen, *Textbook of Drug Design and Discovery* (CRC Press, Florida, 2009).

[81] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui, Molecular docking: A powerful approach for structure-based drug discovery, Curr. Comput. Aided Drug. Des. **7**, 146 (2011).

[82] S. Dastmalchi, M. Hamzeh-Mivehroud, and B. Sokouti, *Methods and Algorithms for Molecular Docking-Based Drug Design and Discovery* (IGI Global, Pennsylvania, 2016).