

A universal DNA computing model for solving NP-hard subset problems

Enqiang Zhu¹, Xianhang Luo¹, Chanjuan Liu²✉, Xiaolong Shi¹✉, and Jin Xu³✉

¹Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China

²School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

³School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

✉ Corresponding authors: chanjuanliu@dlut.edu.cn; xlshi@gzhu.edu.cn; jxu@pku.edu.cn

ABSTRACT

DNA computing, a nontraditional computing mechanism, provides a feasible and effective method for solving **NP**-hard problems because of the vast parallelism and high-density storage of DNA molecules. Although DNA computing has been exploited to solve various intractable computational problems, such as the Hamiltonian path problem, SAT problem, and graph coloring problem, there has been little discussion of designing universal DNA computing-based models, which can solve a class of problems. In this paper, by leveraging the dynamic and enzyme-free properties of DNA strand displacement, we propose a universal model named $\text{DCMS}_{\text{subset}}$ for solving subset problems in graph theory. The model aims to find a minimum (or maximum) set satisfying given constraints. For each element x involved in a given problem, $\text{DCMS}_{\text{subset}}$ uses an exclusive single-stranded DNA molecule to model x as well as a specific DNA complex to model the relationship between x and other elements. Based on the proposed model, we conducted simulation and biochemical experiments on three kinds of subset problems, a minimum dominating set, maximum independent set, and minimum vertex cover. We observed that $\text{DCMS}_{\text{subset}}$ can also be used to solve the graph coloring problem. Moreover, we extended $\text{DCMS}_{\text{subset}}$ to a model for solving the SAT problem. The results of experiments showed the feasibility and universality of the proposed method. Our results highlighted the potential for DNA strand displacement to act as a computation tool to solve **NP**-hard problems.

Keywords: DNA computing; subset problems; **NP**-hard; DNA strand displacement.

Introduction

The existence of **NP**-hard problems that cannot be solved in a polynomial time (unless $\mathbf{P}=\mathbf{NP}$) has inspired researchers to exploit new computing models. DNA computing, as a novel computing model, leverages the Watson–Crick complementary pairing principle and predictable double helical structure, and has been extensively studied by researchers from various areas, including mathematics^{1–3}, computer science^{4,5}, and biology^{6,7}. Original work on DNA computing was undertaken in 1994 by Adelman⁸, who provided an inspiring theory for solving intractable computational problems with biotechnology, and experimentally verified the principle by a simple example of the Hamiltonian path problem. Adelman’s work created a new computing method for solving combinatorial problems, which is called molecular parallelism by Reif⁹. Although scholars have expressed skepticism toward molecular computation owing to its capability to handle only simple problems¹⁰, this argument was quickly destroyed by Lipton¹¹, who demonstrated the feasibility of solving the SAT problem by extending Adelman’s method. Recent studies in^{1,12} have shown that DNA computing can solve the graph coloring problem effectively.

In 1998, to simplify the process of solution detection, Roweis et al.¹³ proposed a sticker-based model for DNA computation, called the sticker model, involving neither enzymes

nor PCR extension. More importantly, the original double-stranded structures can be recovered through memory strands after secondary annealing, which allows for the reuse of the DNA material. Based on the sticker model, in 2002, Zimmermann et al.¹⁴ designed DNA algorithms to solve the counting version of a series of **NP**-hard problems, including k -cliques, independent k -sets, Hamiltonian paths, and Steiner trees. Also in 2002, Braich et al.¹⁵ solved a 20-variable instance of the 3-SAT problem based on the separation operation of the sticker model.

Based on the sticker model, in 2000, Yurke¹⁶ proposed the DNA strand displacement technology, which aims to construct nucleic acid systems with desired dynamic properties. A toehold-mediated strand displacement reaction is described as a molecular dynamic process of replacing a desired single-stranded DNA (called *incumbent*) from a duplex (i.e., a double-stranded DNA with a sticky end) with an input single-stranded DNA (called *invader*) that consists of the desired single-stranded DNA and the complementary strand of the sticky end (i.e., the toehold) to create a more stable complex (a double-stranded DNA)^{17,18}. Figure 1 illustrates the principle of the strand displacement reaction. Compared with the traditional self-assembly models, DNA strand displacement can be completed spontaneously at room temperature. Moreover, the DNA strand displacement is enzyme-free,

which provides with more flexibility in constructing molecular circuits. Significant literature has been published on DNA strand displacement. In 2011, Qian and Winfree⁵ proposed a reversible strand displacement logic gate, called a seesaw gate, based on which they designed a 4-bit square-root circuit containing 130 DNA strands. The seesaw gate has been widely used as a simple building block for constructing large-scale circuits and neural networks⁴. In 2014, Machinek et al.¹⁹ proposed a method of creating mismatched base pairs to achieve the kinetic control of strand displacement. In 2020, Wang et al.²⁰ designed DNA switching circuits based on DNA strand displacement for digital computing. Moreover, Liu et al.¹⁷ proposed a DNA strand displacement circuit called the cross inhibitor, which is time-sensitive and allows interactive inhibition between two input signals. More recently, leveraging the property of programmable interactions between nucleic acid strands, Jung⁶ expanded the capabilities of cell-free biosensors by designing DNA displacement interference circuits. Zhu et al.²¹ first experimentally demonstrated that DNA strand displacement could be applied to encryption.

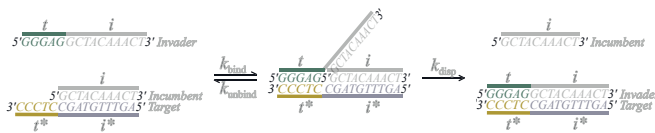


Figure 1. The principle of DNA strand displacement. Initially, an incumbent single-stranded i is hybridized to a complementary domain t^* of a target strand with an extra toehold domain t , forming a duplex. The invader consists of the incumbent single-stranded i and the complementary toehold t of t^* . The displacement starts with the action that the toehold t of the invader binds to t^* and progresses through a branch migration process of replacing the incumbent i by the invader and creating a more stable double-stranded complex. The overall reaction rate is strongly dependent on toehold stability^{17,18}.

Several studies have utilized DNA strand displacement to solve intractable combinatorial optimization problems. In 2018, Tang et al.²² applied DNA strand displacement to solve the 0-1 programming problem, where the model is built based on circular DNA. However, the method is impractical because the number of required species of fluorescence is equal to the number of variables. Therefore, in 2021, the same team² designed a chemical reaction network through three reaction modules (i.e., weighted, sum, and threshold) to solve 0-1 integer programming problems. The reliance on fluorescence was reduced significantly. Also Yang et al.²³ proposed a method to solve the SAT problem, which uses a specific origami structure to represent the solutions and detects the solutions by DNA strand displacement.

Although many models of DNA computation have been proposed to solve various types of problems, many of these models were designed to solve only one type of problem. As a result, more energy and effort are required to design distinct models for different problems. Several attempts have

been made to reduce the operational complexity by simplifying the models, and great processes have been attained^{24,25}. However, different models require different environmental conditions (e.g., temperature, enzyme, and equipment), which hampers the compatibility between models, even though all these models are simplified extremely. This results in a high cost when using DNA computing to solve different problems, which hampers the popularization of molecular computers with powerful capability, like electronic computers.

To deal with the above challenges, researchers have begun to design universal models of DNA computation. In 2019, Woods et al.²⁶ utilized tiles to construct an iterated boolean circuit, which can be used to simulate Turing machines²⁷, universal boolean circuits²⁸, and cellular automata²⁹. In 2020, Wang et al.²⁰ designed a circuit for digital computation by combining DNA strand displacement and switching circuits, which can be applied to a variety of scenarios, such as molecular full adder³⁰ and the 4-bit square-root circuit^{3,31}. In 2022, Xie et al.³² designed a three-way junction-incorporated double hairpin unit by combining single-stranded gates³³ and exponential amplification reaction³⁴, which can achieve multiple functions and applications, including a 4-to-2 encoder³⁵, 1-to-2 demultiplexer³⁶, 1-to-4 demultiplexer³⁷, and multi-input OR gate.

However, regarding NP-hard problems, previous DNA computing models are still problem dependent, making the development of universal DNA computing models fascinating but challenging. Among NP-hard problems, there are many classic problems, such as the minimum dominating set, the maximum independent set, and the minimum vertex covering, which aim to find minimum or maximum subsets that satisfy some given restricted conditions. These problems belong to the subset problem in graph theory. This paper proposes a universal DNA computing model named DCMSubset for solving these problems. Given a vertex, DCMSubset utilizes the Watson–Crick complementary pairing principle to model the relationship of the vertex by a specific DNA complex. The set of such DNA complexes corresponding to all vertices forms the computation gate circuit of DCMSubset. Different elements (vertex or edge) are represented by different single-stranded sequences. Thus, each element is equipped with a separate detection gate that uses DNA strand displacement reactions to accurately detect the element. To demonstrate the feasibility of DCMSubset, we conducted simulation and biochemical experiments on the above three subset problems, the results of which show the universality of DCMSubset. We observed that graph coloring could also be solved by DCMSubset, which is a direct application of the maximum independent set. Moreover, we extended DCMSubset to a suitable model for solving the SAT problem. The results suggest the potential of DCMSubset to solve a wider range of NP-hard problems.

Results

The principle of DCMSubset. Regarding subset problems,

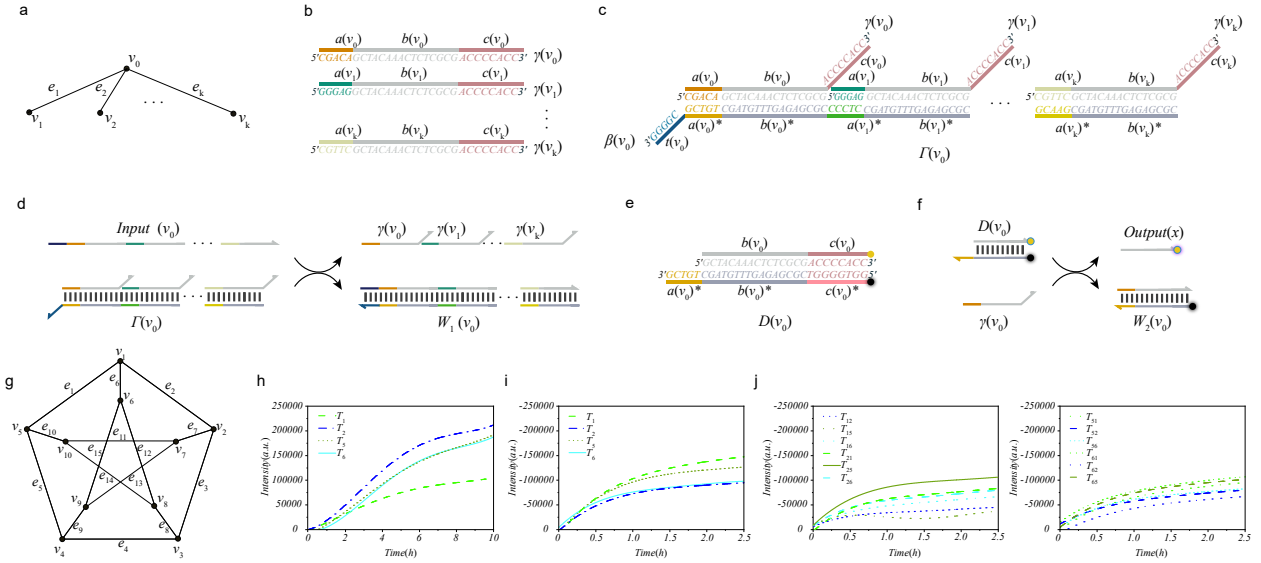


Figure 2. Principle of DCMSubset. **a** A star graph S_k . **b** The structure of single strands $\gamma(v_i)$, $i = 0, 1, \dots, k$, representing the vertices of S_k . **c** The structure of the computation gate $\Gamma(v_0)$, representing the adjacent relation between v_0 and other vertices of S_k . **d** The strand displacement reaction of $\Gamma(v_0)$ and the input single-stranded $Input(v_0)$, where $Input(v_0)$ is the complementary strand of $\beta(v_0)$. **e** The structure of detection gate $D(v_0)$ of the vertex v_0 of S_k . **f** The strand displacement reaction of $D(v_0)$ and $\gamma(v_0)$. **g** The Peterson graph. **h** Reaction kinetics of the experiment showing the feasibility of DCMSubset. **i** Reaction kinetics of the first kind of possible leakage reactions. **j** Reaction kinetics of the second kind of possible leakage reactions. Where the curves T_i (in Fig. 2 h and i) and T_{ij} (in Fig. 2 j) for $i, j \in \{1, 2, 5, 6\}$ and $i \neq j$ were plotted by transferring the cycle value into the reaction time, indicating the change of the relative fluorescence unit (RFU) values in the FAM channel.

modeling elements and their relations using DNA molecules is fundamental to constructing universal models. This paper focuses on problems related to graphs, where a graph G is often described as a 2-tuple (V, E) such that $V (\neq \emptyset)$ represents the *vertex set* of G , and E (a set of two-element subsets of V) represents the *edge set* of G . Two vertices are *adjacent* if and only if they are the ends of an edge of G , while a vertex is *incident* with an edge if and only if the vertex is an end of the edge. Given a vertex u , the set $N(u)$ of vertices adjacent to u is called the *neighborhood* of u , and $N[u] = N(u) \cup \{u\}$ is called the *close neighborhood* of u ; similarly, the set $N_e(u)$ of edges incident to u is called the *edge-neighborhood* of u , and $N_e[u] = N_e(u) \cup \{u\}$ is the *close edge-neighborhood* of u .

The problem we consider in this paper, named p -SUBSET $[\phi]$, can be formally described as follows.

Problem 1 p -SUBSET $[\phi]$: Given a graph $G = (V, E)$ and a property ϕ , the problem asks to find a subset S of V with the minimum (or maximum) cardinality such that S satisfies ϕ .

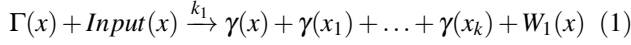
Below is an account of the design of DCMSubset for solving p -SUBSET $[\phi]$ problems.

Based on ϕ , we first determine the underlying set U of the problem; generally, $U = V$ or $U = V \cup E$. For each element $x_0 \in U$, we design a 29-nt single-stranded DNA (denoted by $\gamma(x_0)$) to represent it and use a DNA complex (denoted by $\Gamma(x_0)$) to characterize a relation (that is relevant to ϕ , e.g., the

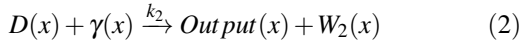
adjacent relation and the incident relation) between x_0 and other elements. $\gamma(x_0)$ consists of a 5-nt toehold domain $a(x_0)$, a 16-nt branch migration domain $b(x_0)$, and a 8-nt branch migration domain $c(x_0)$. $\Gamma(x_0)$ is obtained by binding all $\gamma(x_i)$ s ($i = 0, 1, \dots, k$) to a long single-stranded (denoted by $\beta(x_0)$) that consists of a 5-nt toehold $t(x_0)$ at its 3'-end and the complementary strands $a(x_j)^*$ and $b(x_j)^*$ of $a(x_j)$ and $b(x_j)$, respectively, for $j \in \{0, \dots, k\}$, where x_j is either x_0 or an element associated with x_0 . All $\Gamma(x)$ s for $x \in U$ build the computation gate circuit of DCMSubset. To illustrate this modeling method, we consider the star graph S_k , shown in Fig. 2 a, where the vertex set $V = \{v_i | i = 0, 1, \dots, k\}$ and the edge set $E = \{e_i = v_0 v_i | i = 1, 2, \dots, k\}$. We, for example, consider the adjacent relation and $U = V$. For $i = 0, 1, \dots, k$, v_i is modeled as a 29-nt single-stranded $\gamma(v_i)$ (see Fig. 2 b) and the close neighborhood of v_i is modeled as a duplex $\Gamma(v_i)$ (see Fig. 2 c for an illustration of $\Gamma(v_0)$).

Each computation gate corresponding to an element $x \in U$ receives an exclusive input signal strand $Input(x)$ (the complementary strand of $\beta(x)$) as the invader to displace the incumbent strands, that is, $\gamma(x)$ and $\gamma(x_i)$ for $i = 1, \dots, k$, where x_i is an element associated with x . See Fig. 2 d for the strand displacement reaction of $\Gamma(v_0)$ and $Input(v_0)$, which releases $\gamma(v_i)$, $i = 0, 1, \dots, k$ and a stable double strand ($W_1(v_0)$). The computation gate mechanism can be expressed via the follow-

ing reaction:



To detect the released single strands by the computation gate circuit, we need a detection gate for each element to recognize the released single strands. For each element x , the detection gate of x , denoted by $D(x)$, is a duplex that binds an incumbent strand consisting of $b(x)$ and $c(x)$ to a target strand consisting of $a(x)^*$, $b(x)^*$ and $c(x)^*$. Here, the incumbent strand carries a fluorophore at its 3'-end, and the target strand carries a quencher at its 5'-end. Because the quencher is close to the fluorophore, the fluorescence is inhibited in $D(x)$. When the invader $\gamma(x)$ exists, its toehold $a(x)$ binds to the domain $a(x)^*$ of $D(x)$, and branch migration moves gradually to the domain $c(x)^*$, which releases a single-stranded $Output(x)$ with a fluorescence signal for monitoring and a stable beacon-labeled double-stranded $W_2(x)$. See Fig. 2 e and f for the structure of $D(v_0)$ and the strand displacement reaction between $D(v_0)$ and $\gamma(v_0)$, respectively. All $D(x)$ s build the detection gate circuit of DCMSubset for $x \in U$. The detection gate mechanism can be expressed via the following reaction:



DCMSubset consists of computation gate circuit and the detection gate circuit, which work as follows.

Regarding a p -SUBSET[ϕ] problem \mathcal{P} on the underlying set $U = \{v_1, v_2, \dots, v_n\}$, for the purpose of finding a minimum (or maximum) subset satisfying ϕ , we have to traverse over all the subsets of U . To determine whether a subset $U' = \{u_1, u_2, \dots, u_\ell\} \subseteq U$ satisfies ϕ , we first add the computation gates $\Gamma(u_1), \Gamma(u_2), \dots, \Gamma(u_\ell)$ into a tube T and then divide T equally into n tubes T_1, T_2, \dots, T_n . Second, we add the detection gates $D(u_i)$ into the i th tube T_i for $i = 1, 2, \dots, n$. Third, we add all $Input(u_i)$, $i = 1, 2, \dots, \ell$, to each T_j for $j \in \{1, 2, \dots, n\}$ and detect the fluorescence signals. It should be noted that computation gates should be designed in compliance with the property ϕ ; that is, the design should ensure that U' is a solution of \mathcal{P} if and only if the combination of fluorescence signals in all T_i ($i = 1, 2, \dots, \ell$) is a certificate of the property ϕ (e.g., the fluorescence signals in all tubes are detected or none of the tubes are detected).

Feasibility of DCMSubset. To verify the feasibility of DCMSubset, we conducted a biochemical experiment for detecting the close neighborhood $N[v_1] = \{v_1, v_2, v_5, v_6\}$ of the vertex v_1 in the Peterson graph \mathbf{P} (as shown in Fig. 2 g). We first designed the input strands $Input(v_1)$, computation gate $\Gamma(v_1)$, and detection gates $D(v_1), D(v_2), D(v_5)$, and $D(v_6)$; see Supplementary Table S1 and S2 for the details of these strands. To verify that $\Gamma(v_1)$ can release single strands $\gamma(v_1), \gamma(v_2), \gamma(v_5)$ and $\gamma(v_6)$, we found it sufficient to prepare four tubes (say T_1, T_2, T_5 , and T_6) such that T_i contains the mixture of $\Gamma(v_1)$ and $D(v_i)$ for $i = 1, 2, 5, 6$. For each T_i , we first added $D(v_i)$ ($10 \mu\text{l} \times 1 \mu\text{M}$) and the computation gates $\Gamma(v_1)$ ($10 \mu\text{l} \times 2 \mu\text{M}$); then, we added $Input(v_1)$ ($10 \mu\text{l} \times$

$1 \mu\text{M}$); finally, all T_i were put into a fluorescence quantification PCR machine for detecting the fluorescence output. The experimental results are shown in Fig. 2 h, from which we can see that the fluorescence signals in all these tubes are captured.

In addition, we can observe that the strand displacement reaction for $D(v_i)$ (Eq. (2)) is triggered by the toehold domain $a(v_i)^*$ and proceeds from the branch migration domains $b(v_i)^*$ and $c(v_i)^*$; the input strand $Input(v_i)$ for $\Gamma(v_i)$ (Eq. (1)) contains $a(v_i)$ and $b(v_i)$; $\gamma(v_i)$ contains $a(v_i)$, $b(v_i)$, and $c(v_i)$. Therefore, two kinds of experiments were carried out to analyze the possible leakage reactions: (1) the reactions of $Input(v_1)$ and $D(v_j)$ for $j = 1, 2, 5, 6$; and (2) the reactions of $\gamma(v_j)$ and $D(v_k)$ for $j, k \in \{1, 2, 5, 6\}$ and $j \neq k$.

Regarding the first kind of experiment, we prepared four tubes T_j ($j \in \{1, 2, 5, 6\}$). For each tube T_j , we added the mixture of $D(v_j)$ ($15 \mu\text{l} \times 0.67 \mu\text{M}$) and $Input(v_1)$ ($15 \mu\text{l} \times 0.67 \mu\text{M}$). Regarding the second experiment, we prepared 12 tubes denoted by T_{jk} ($j, k \in \{1, 2, 5, 6\}$ and $j \neq k$), to which we added $\gamma(v_i)$ ($15 \mu\text{l} \times 1.33 \mu\text{M}$) and $D(v_j)$ ($15 \mu\text{l} \times 0.67 \mu\text{M}$). To monitor the fluorescence signals, we put all the tubes T_i and T_{ij} into the fluorescence quantification PCR machine. It can be seen that no fluorescence signal is detected in all these tubes (see Fig. 2 i and j). This indicates that the possible leakage reactions in DCMSubset do not affect the accuracy of the detection gate.

Below is the applications of DCMSubset to three important subset problems in graph theory: the minimum dominating set, the maximum independent set, and the minimum vertex cover. All graphs considered in the paper contain no loop or parallel edge, i.e., simple graphs.

Application to minimum dominating set. Given a graph $G = (V, E)$, a *dominating set* (DS) of G is a subset S of V such that every vertex in $V \setminus S$ is adjacent to a vertex in S .

Problem 2 *The minimum dominating set (MDS) problem requires finding a DS of the minimum cardinality.*

We used DCMSubset to find an MDS of the Peterson graph \mathbf{P} shown in Fig. 2 g. In order to reduce costs, we first dealt with the solution space by some preprocessing with a theoretical guarantee. We needed the following two results, where Proposition 1 follows directly from the definition of DS, and Proposition 2 was obtained by Bruce³⁸ in 1996.

Proposition 1 *If S is a DS of a graph $G = (V, E)$, then $|N[S]| \geq |V|$, where $N[S] = \cup_{v \in S} N[v]$.*

Proposition 2 *Let S be an MDS of a graph such that every vertex is adjacent to at least three vertices. Then, $|S| \leq \frac{3|V|}{8}$.*

As $|N[v]| = 4$ for any vertex of \mathbf{P} , it has $|N[S]| \leq 8$ for any $S \subset V$ with $|S| = 2$. Thus, according to Proposition 1, any dominating set of \mathbf{P} contains at least three vertices. Moreover, by Proposition 2, we know that any MDS of \mathbf{P} contains exactly three vertices. Therefore, the solution space Ω is the set of all 3-subsets with cardinality $\binom{10}{3} = 120$; that is,

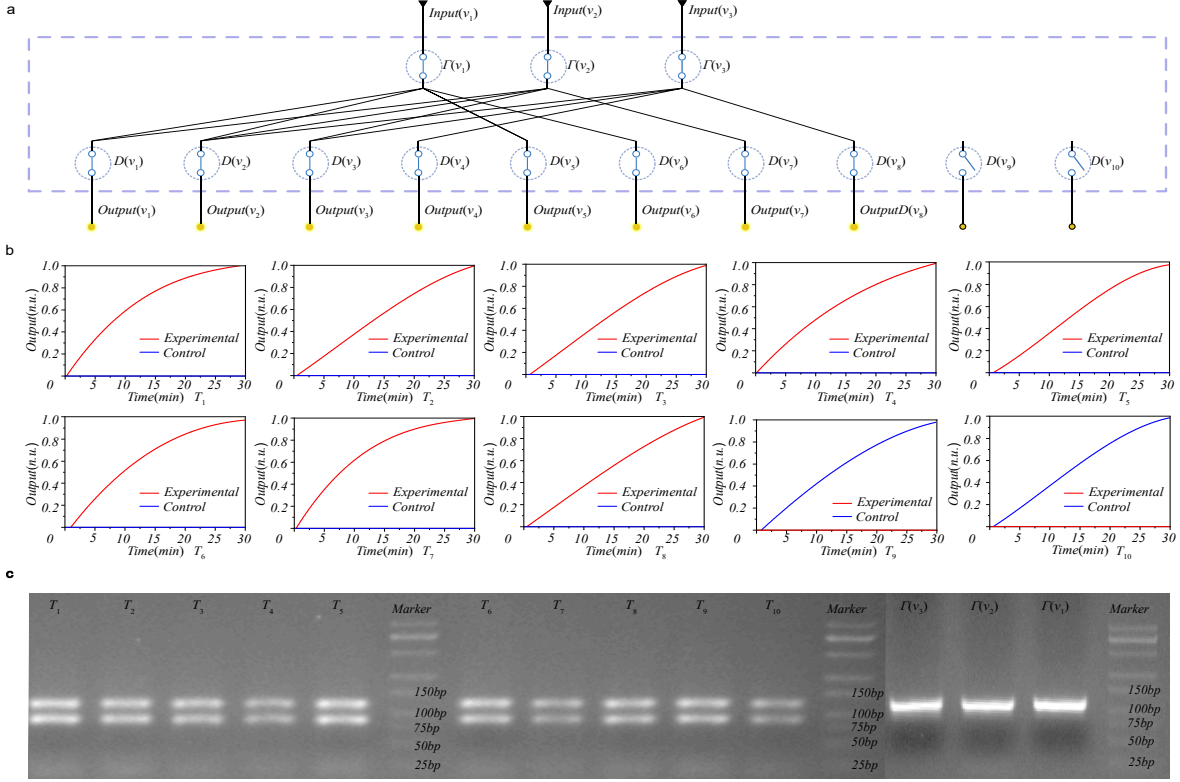


Figure 3. Experiments for determining whether $\{v_1, v_2, v_3\}$ is a DS of \mathbf{P} . **a** Logic circuit implementation of the first group of experiments. **b** Reaction kinetics of the first group of experiments, which corresponds to the input $\{v_1, v_2, v_3\}$. The curve was plotted by transferring the cycle value to the reaction time. The outputs were normalized to RFU values in the FAM channel with the highest signals. **c** Agarose gel electrophoresis results, where for $i = 1, 2, \dots, 10$, T_i contains input strands ($Input(v_1)$, $Input(v_2)$, and $Input(v_3)$), computation gates ($\Gamma(v_1)$, $\Gamma(v_2)$, and $\Gamma(v_3)$), and the detection gate $D(v_i)$.

$\Omega = \{\{v_1, v_2, v_3\}, \{v_1, v_2, v_4\}, \dots, \{v_8, v_9, v_{10}\}\}$. For description convenience, we used S_i , $i = 1, 2, \dots, 120$ to denote the 120 candidate solutions in Ω .

We now focus on finding an MDS of \mathbf{P} using DCMSubset. As previously mentioned, for $i = 1, 2, \dots, 10$, v_i is modeled as a single-stranded $\gamma(v_i)$ that consists of three parts: $a(v_i)$, $b(v_i)$, and $c(v_i)$. The adjacent relation of v_i is modeled as a duplex $\Gamma(v_i)$ (i.e., the computation gate of v_i), which is hybridized by $\gamma(x)$ for all $x \in N[v_i]$ and a long single-stranded $\beta(v_i)$, where $\beta(v_i)$ consists of a 5-nt toehold $t(v_i)$ and the complementary strands of $a(x)$ and $b(x)$. The detection gate $D(v_i)$ is a duplex that is hybridized by a single strand consisting of $b(v_i)$ and $c(v_i)$ and a single strand consisting of the complementary strands of $a(v_i)^*$, $b(v_i)^*$, and $c(v_i)^*$. See Supplementary Tables S1 and S2 for the details of these strands.

The experiments are described as follows. We prepared 120 groups of experiments for testing the solution space, where each group consists of 10 tubes. The i th group of experiments was used to determine whether S_i is an MDS of \mathbf{P} . We use T_{ij} to denote the j th tube in the i th group, where $i = 1, 2, \dots, 120$, $j = 1, 2, \dots, 10$. We first added $D(v_j)$ (12.5

$\mu\text{l} \times 1.1 \mu\text{M}$) and the three computation gates $\Gamma(x)$, $\Gamma(y)$, and $\Gamma(z)$ (each $4 \mu\text{l} \times 5.5 \mu\text{M}$) corresponding to the three vertices in $S_i = \{x, y, z\}$ to each tube T_{ij} . Then, we added $Input(x)$, $Input(y)$, and $Input(z)$ (each $2.5 \mu\text{l} \times 5.5 \mu\text{M}$) to T_{ij} . Finally, all tubes T_{ij} were put into the fluorescence quantification PCR machine for detecting the fluorescence output. See Fig. 3 for the diagram of the first group of experiments, which corresponds the candidate solution $S_1 = \{v_1, v_2, v_3\}$.

According to the above discussion, a vertex $s \in N[s']$ for some $s' \in \{x, y, z\}$ implies that s must be released by the displacement reactions of $\Gamma(s')$ and $Input(s')$. Therefore,

Theorem 1 *The i th candidate solution $S_i \in \Omega$ is a DS of \mathbf{P} if and only if the fluorescence signal in T_{ij} is detected for all $j \in \{1, 2, \dots, 10\}$.*

Here, we emphasize that we stopped our experiments as soon as a solution was found, and all groups of experiments were conducted similarly (if all the 120 groups of experiments were implemented, then all of the MDS of \mathbf{P} could be found). Thus, only 12 groups of experiments were implemented, corresponding to the inputs (candidate solutions) $S_1 = \{v_1, v_2, v_3\}$, $S_2 = \{v_1, v_2, v_4\}$, $S_3 = \{v_1, v_2, v_5\}$,

$S_4 = \{v_1, v_2, v_6\}$, $S_5 = \{v_1, v_2, v_7\}$, $S_6 = \{v_1, v_2, v_8\}$, $S_7 = \{v_1, v_2, v_9\}$, $S_8 = \{v_1, v_2, v_{10}\}$, $S_9 = \{v_1, v_3, v_4\}$, $S_{10} = \{v_1, v_3, v_5\}$, $S_{11} = \{v_1, v_3, v_6\}$, and $S_{12} = \{v_1, v_3, v_7\}$. By Theorem 1, S_i , $i = 1, 2, \dots, 12$ is a DS if all of the 10 tubes of the i th group of experiments output fluorescence signals. Figure 3 b shows the result of fluorescence detection of the first group of experiments, in which the ninth and tenth tubes do not yield a fluorescence signal, which implies that S_1 is not a DS of \mathbf{P} . The results of fluorescence detection of the other 11 groups of experiments are shown in Supplementary Fig. S1–S3, from which we see that S_{12} is a DS of \mathbf{P} and also an MDS of \mathbf{P} .

To test that the experiments can produce the desired outcomes (double-stranded W_1 in Eq. (1)), we separately carried out agarose gel electrophoresis experiments for pure $\Gamma(x)$ and the mixture of $\Gamma(x)$, $Input(x)$, and $D(x)$. Figure 3 c shows the experimental results corresponding to the three inputs $Input(v_1)$, $Input(v_2)$, and $Input(v_3)$ (the results of the other 11 groups of experiments are shown in Supplementary Fig. S3), from which we see that a new band between 75 bp and 100 bp is produced in the mixture solution, while no new band is produced in the pure $\Gamma(x)$ solution. This result is consistent with the design of $DCMS_{\text{subset}}$ because the double-stranded structure $W_1(x)$ formed by the strand displacement reaction of $\Gamma(x)$ and $Input(x)$ (Eq. (1)) has the length 89 bp.

Application to maximum independent set. Given a graph $G = (V, E)$, an *independent set* (IS) of G is a subset $S \subseteq V$ such that any two vertices in S are not adjacent.

Problem 3 *The maximum independent set (MIS) problem requires finding an IS of the maximum cardinality.*

We used $DCMS_{\text{subset}}$ to find an MIS of Peterson graph \mathbf{P} (see Fig. 2 h). Analogously, we first reduced the solution space by some preprocessing with theoretical guarantee. We utilized the following two results: Proposition 3 follows directly by the pigeonhole principle and the fact that the set of vertices with the same color is an independent set; Proposition 4 was obtained by Lovász in 1979³⁹.

Proposition 3 *Let I be an IS of an n -vertex graph G . Then, $|I| \geq \lfloor \frac{n}{\chi(G)} \rfloor$, where $\chi(G)$ is the chromatic number of G , that is, the minimum number of colors assigned to the vertices of G such that no two adjacent vertices receive the same color.*

Proposition 4 *Let I be an MIS of an n -vertex graph G . Then, $|I| \leq \lceil \frac{-n\lambda_{\min}(A)}{\lambda_{\max}(A) - \lambda_{\min}(A)} \rceil$, where A is the adjacency matrix of G , and $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the maximum and minimum eigenvalues of A , respectively.*

As the chromatic number of a graph that is neither isomorphic to K_4 (the complete graph on four vertices) nor an odd cycle is at most the maximum degree⁴⁰, it follows that $\chi(\mathbf{P}) \leq 3$, so any MIS of \mathbf{P} contains at least four vertices, according to Proposition 3. Additionally, the maximum and minimum eigenvalues of the adjacency matrix of \mathbf{P} are 3 and -2, respectively, which implies that an MIS of \mathbf{P} contains at

most four vertices, according to Proposition 4. Therefore, an MIS of \mathbf{P} contains exactly four vertices, and the solution space Ω is the set of all 4-subsets with the cardinality $\binom{10}{4}=210$; that is, $\Omega = \{\{v_1, v_2, v_3, v_4\}, \{v_1, v_2, v_3, v_5\}, \dots, \{v_7, v_8, v_9, v_{10}\}\}$. We use I_k , $k = 1, 2, \dots, 210$ to denote the 210 candidate solutions in Ω .

We now focus on the experiment of finding an MIS of \mathbf{P} base on $DCMS_{\text{subset}}$. All key components were designed in a manner similar to the MDS problem except for a slight change to the computation gate $\Gamma(v_i)$, $i = 1, 2, \dots, 10$. Here, each $\Gamma(v_i)$ was modified by just deleting the $c(v_i)$ component of $\gamma(v_i)$ at the 5'-end, and the other components were unchanged. We denote $\Gamma'(v_i)$ as the new computation gate; see Fig. 4 a for $\Gamma'(v_0)$. The experiments were conducted as follows. We prepared 210 groups of experiments for testing the solution space, where each group used one tube. The k th group of experiments (denoted by T_k) aimed to determine whether I_k is an MIS of \mathbf{P} . Suppose that $I_k = \{x, y, z, w\}$. We first added the four detection gates $D(x)$, $D(y)$, $D(z)$, $D(w)$ (each $5 \mu\text{l} \times 0.916 \mu\text{M}$) and the four computation gates $\Gamma'(x)$, $\Gamma'(y)$, $\Gamma'(z)$ and $\Gamma'(w)$ (each $5 \mu\text{l} \times 1.832 \mu\text{M}$) to T_k . Then, we added $Input(x)$, $Input(y)$, $Input(z)$ and $Input(w)$ (each $5 \mu\text{l} \times 0.916 \mu\text{M}$) to T_k . Finally, T_k was put into a fluorescence quantification PCR machine for detecting the fluorescence output. See Fig. 4 c for the circuit diagram of the first group of experiments, which corresponds to the solutions $I_1 = \{v_1, v_2, v_3, v_4\}$.

It is clear that if an arbitrary vertex $s \in \{x, y, z, w\}$ is adjacent to one of x, y, z , and w , then $\gamma(s)$ is released by $\Gamma'(s')$ for some $s' \in \{x, y, z, w\}$ and then detected by $D(s)$; moreover, if $\{x, y, z, w\}$ is an IS, then no $\gamma(s)$ for $s \in \{x, y, z, w\}$ is released by $\Gamma'(s')$ for any $s' \in \{x, y, z, w\}$. We can observe that $\Gamma'(s)$ and $Input(s)$ can only produce a partial strand of $\gamma(s)$ that consists of only $a(s)$ and $b(s)$, denoted by $\gamma'(s)$. Also, $\gamma'(s)$ and $D(s)$ do not produce fluorescence signals (see Fig. 4 b). Therefore, the following result holds.

Theorem 2 *The k th candidate solution $I_k \in \Omega$ is an IS of \mathbf{P} if and only if no fluorescence signal is detected in T_k .*

Also, we stopped our experiments as soon as a solution was found. Thus, we only conducted 49 groups of experiments, corresponding to the candidate solutions $\{v_1, v_2, v_3, v_4\}$, $\{v_1, v_2, v_3, v_5\}$, $\{v_1, v_2, v_3, v_6\}$, $\{v_1, v_2, v_3, v_7\}$, $\{v_1, v_2, v_3, v_8\}$, $\{v_1, v_2, v_3, v_9\}$, $\{v_1, v_2, v_3, v_{10}\}$, $\{v_1, v_2, v_4, v_5\}$, $\{v_1, v_2, v_4, v_6\}$, $\{v_1, v_2, v_4, v_7\}$, $\{v_1, v_2, v_4, v_8\}$, $\{v_1, v_2, v_4, v_9\}$, $\{v_1, v_2, v_4, v_{10}\}$, $\{v_1, v_2, v_5, v_6\}$, $\{v_1, v_2, v_5, v_7\}$, $\{v_1, v_2, v_5, v_8\}$, $\{v_1, v_2, v_5, v_9\}$, $\{v_1, v_2, v_5, v_{10}\}$, $\{v_1, v_2, v_6, v_7\}$, $\{v_1, v_2, v_6, v_8\}$, $\{v_1, v_2, v_6, v_9\}$, $\{v_1, v_2, v_6, v_{10}\}$, $\{v_1, v_2, v_7, v_8\}$, $\{v_1, v_2, v_7, v_9\}$, $\{v_1, v_2, v_7, v_{10}\}$, $\{v_1, v_2, v_8, v_9\}$, $\{v_1, v_2, v_8, v_{10}\}$, $\{v_1, v_2, v_9, v_{10}\}$, $\{v_1, v_3, v_4, v_5\}$, $\{v_1, v_3, v_4, v_6\}$, $\{v_1, v_3, v_4, v_7\}$, $\{v_1, v_3, v_4, v_8\}$, $\{v_1, v_3, v_4, v_9\}$, $\{v_1, v_3, v_4, v_{10}\}$, $\{v_1, v_3, v_5, v_6\}$, $\{v_1, v_3, v_5, v_7\}$, $\{v_1, v_3, v_5, v_8\}$, $\{v_1, v_3, v_5, v_9\}$, $\{v_1, v_3, v_5, v_{10}\}$, $\{v_1, v_3, v_6, v_7\}$, $\{v_1, v_3, v_6, v_8\}$, $\{v_1, v_3, v_6, v_9\}$, $\{v_1, v_3, v_6, v_{10}\}$, $\{v_1, v_3, v_7, v_8\}$, $\{v_1, v_3, v_7, v_9\}$, $\{v_1, v_3, v_7, v_{10}\}$, $\{v_1, v_3, v_8, v_9\}$, $\{v_1, v_3, v_8, v_{10}\}$, and $\{v_1, v_3, v_9, v_{10}\}$, where $\{v_1, v_3, v_9, v_{10}\}$ is an MIS, while the

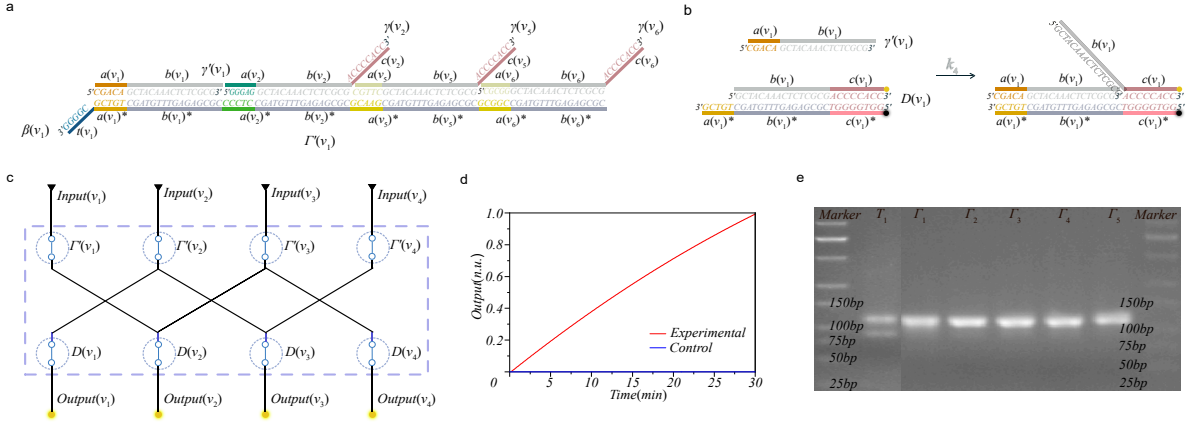


Figure 4. The first group of experiments that determine whether $\{v_1, v_2, v_3, v_4\}$ is an IS of \mathbf{P} . **a** The structure of computation gate $\Gamma'(v_1)$. **b** The strand displacement reaction of $\gamma(v_1)$ and the detection gate $D(v_1)$, where the incumbent strand of $D(v_1)$ (consists of domains $b(v_1)$ and $c(v_1)$) cannot be released. **c** Logic circuit implementation. **d** Reaction kinetics of the experiments, where the curve was plotted by transferring the cycle value into the reaction time, and the outputs were normalized to the RFU values in the FAM channel with the highest signals. **e** Agarose gel electrophoresis results, where T_1 contains input strands ($Input(v_1), Input(v_2), Input(v_3)$, and $Input(v_4)$), computation gates ($\Gamma'(v_1), \Gamma'(v_2), \Gamma'(v_3)$ and $\Gamma'(v_4)$), and detection gates ($D(v_1), D(v_2), D(v_3), D(v_4)$).

others are not ISs of \mathbf{P} . Figure 4 d shows the result of fluorescence detection of the first group of experiments, which implies that S_1 is not an IS of \mathbf{P} . Figure 4 e shows the result of agarose gel electrophoresis of the first group of experiments. The results of the other 48 groups of experiments are shown in Supplementary Figs. S4–S5, from which we see that S_{49} is an IS of \mathbf{P} and an MIS of \mathbf{P} .

As a direct extension, we observe that the DCMSubset circuit can be used to solve the k -coloring problem, which requires us to determine whether there exists an assignment of k colors to the vertices of a graph G such that no two adjacent vertices receive the same color. As such an assignment is a k -coloring of G if and only if every set of vertices that receive the same color is an IS of G , it is enough to enumerate all k -partitions of $V(G)$ (i.e., divide $V(G)$ into k disjoint subsets), and for each k -partition V_1, V_2, \dots, V_k (where $V_1 \cup V_2 \cup \dots \cup V_k = V(G)$ and $V_i \cap V_j = \emptyset$ for $i, j \in \{1, 2, \dots, k\}$ and $i \neq j$), we can determine whether V_i (for $i = 1, 2, \dots, k$) is an IS of G using the DCMSubset circuit for the MIS problem. If V_i is an IS for all $i \in \{1, 2, \dots, k\}$, then V_1, V_2, \dots, V_k corresponds to a k -coloring (vertices in V_i is colored with i); otherwise, V_1, V_2, \dots, V_k does not correspond to a k -coloring, and we consider the next k -partition.

Below we argue the applications of DCMSubset to the minimum vertex cover problem. As the experiments were implemented in the same way as the MDS problem, we only verified the correctness of the designed circuits by simulation experiments.

Application to minimum vertex coverage. Given a graph $G = (V, E)$, a *vertex covering* (VC) of G is a subset $C \subseteq V$ such that each edge is incident with at least one vertex of C .

Problem 4 The minimum vertex covering (MVC) problem requires us to find a VC of the minimum cardinality.

Next, we constructed the DCMSubset circuit for finding all MVCs of the Peterson graph \mathbf{P} . Because an MVC involves both vertices and edges of a graph, we had to model both of these two types of elements. While all of the key components of the circuit were designed in the same way as that for the MDS problem, single-stranded $\gamma(e_i)$ and complex $D(e_i)$, for $i = 1, 2, \dots, 15$ were added to model the edges of \mathbf{P} and their detection gate. Moreover, each computation gate $\Gamma(v_i)$, $i = 1, 2, \dots, 10$, indicating the incident relation, was hybridized by $\gamma(x)$ for $x \in \{v_i, e_{i_1}, e_{i_2}, e_{i_3}\}$ and a long single-stranded $\beta(v_i)$ consisting of a 5-nt toehold $t(v_i)$ and the complementary domains of $a(x)$ and $b(x)$. Here, e_{i_1}, e_{i_2} , and e_{i_3} are the three edges that are incident with v_i . See Supplementary Fig. S8 for the illustration of $\Gamma(v_1)$.

For an n -vertex graph G , it is well known that $|I| + |C| = n$ (first observed by Gallai⁴¹ in 1959), where I and C are an MIS and MVC of G . Therefore, an MCV of \mathbf{P} contains six vertices, and the solution space Ω is the set of all 6-subsets of $V(\mathbf{P})$ with the cardinality $\binom{10}{6} = 210$. Let $\Omega = \{C_i | i = 1, 2, \dots, 210\}$.

The simulation experiments were implemented as follows. We prepared 210 groups of experiments. For each $i = 1, 2, \dots, 210$, the i th group of experiments involves 15 specific experiments, for the purpose of determining whether C_i is an MVC of \mathbf{P} . We use T_{ij} to denote the j th experiment in i th group, where $j = 1, 2, \dots, 15$. Let $C_i = \{v_{i_k} | k = 1, 2, \dots, 6\}$. For each group T_{ij} , we first added $D(e_j)$ ($10nM$), then the six computation gates $\Gamma(x_{i_k})$ (each $20nM$), and finally the six inputs $Input(x_{i_k})$ (each $10nM$), where $k = 1, 2, \dots, 6$. To assess whether C_i is a VC, we detected the concentration of the output signal using DSD simulation tool. See Fig. 5 for the

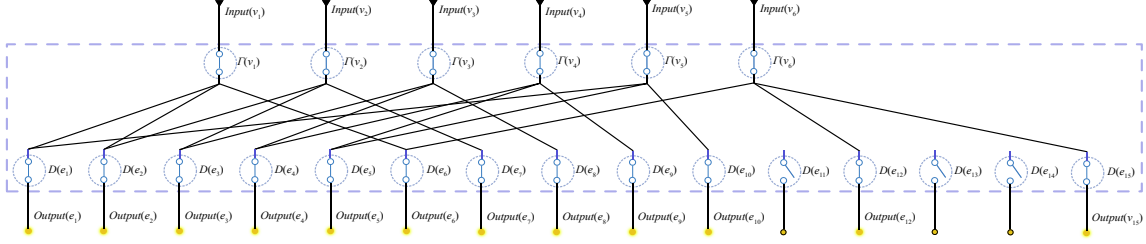


Figure 5. Logic circuit implementation of determining whether or not $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ is a VC of \mathbf{P} based on DCMSubset.

circuit of the first group of experiments, which corresponds to the inputs $Input(v_k)$ for $k = 1, 2, \dots, 6$.

Through the above analysis, we obtained the following result.

Theorem 3 $C_i \in \Omega$ is a VC of \mathbf{P} if and only if the concentration of the output signal in T_{ij} is detected for all $j \in \{1, 2, \dots, 15\}$.

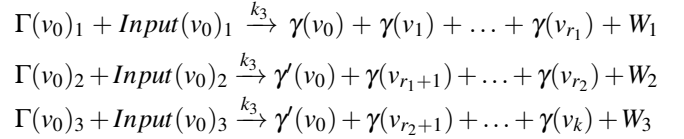
We conducted 210 groups of simulation experiments, corresponding to the 210 candidate solutions $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, $\{v_1, v_2, v_3, v_4, v_5, v_7\}$, \dots , $\{v_5, v_6, v_7, v_8, v_9, v_{10}\}$, where $\{v_1, v_2, v_4, v_8, v_9, v_{10}\}$, $\{v_1, v_3, v_4, v_6, v_7, v_{10}\}$, $\{v_1, v_3, v_5, v_7, v_8, v_9\}$, $\{v_2, v_3, v_5, v_6, v_9, v_{10}\}$, and $\{v_2, v_4, v_5, v_6, v_7, v_8\}$ are MVCs. The simulation results are illustrated in Supplementary Figs. S11–S17.

Discussion

We demonstrated that DCMSubset could be applied to subset problems in graph theory. In contrast with previous DNA computing models, DCMSubset is more general.

Splitting computation gates. We observed that when a vertex x had a big neighborhood, the single-stranded $\beta(x)$ in the computation gate $\Gamma(x)$ was significantly longer. This may increase the likelihood of error in the process of molecular assembly. We attempted to solve this problem by a splitting method with regard to computation gates. Consider the computation gate $\Gamma(v_0)$ shown in Fig. 2 c, which is obtained by binding $\gamma(v_i)$, $i = 0, 1, \dots, k$ to a single-stranded $\beta(v_0)$. It is clear that the length of $\beta(v_0)$ increases with the increase of k . We controlled the length of $\beta(v_0)$ by splitting $\{v_0, v_1, \dots, v_k\}$ into ℓ subsets with a similar cardinality. We took $\ell = 3$ as an example to show the splitting method, and other cases could be dealt with similarly. First, construct a covering of $\{v_0, v_1, \dots, v_k\}$ that consists of three subsets $N_1 = \{v_0, v_1, \dots, v_{r_1}\}$, $N_2 = \{v_0, v_{r_1+1}, v_{r_1+2}, \dots, v_{r_2}\}$, and $N_3 = \{v_0, v_{r_2+1}, v_{r_1+r_2+2}, \dots, v_k\}$, where $r_1 = \lfloor \frac{k}{3} \rfloor$ and $r_2 = r_1 + \lfloor \frac{k-r_1}{2} \rfloor$. Then, the computational gate $\Gamma(v_0)$ (see Supplementary Fig. S10 a) of v_0 is split into three components $\Gamma(v_0)_1$ (see Supplementary Fig. S10 b), $\Gamma(v_0)_2$ (see Supplementary Fig. S10 c), and $\Gamma(v_0)_3$ (see Supplementary Fig. S10 d), where for $i = 1, 2, 3$, $\Gamma(v_0)_i$ is the duplex hybridized by all $\gamma(x)$ for $x \in N_i$ and a single-stranded $\beta_i(v_0)$ that consists of

a 5-nt toehold t at its 3'-end and the complementary strands of $a(x)$ and $b(x)$ for $x \in N_i$. Here, a slight change to $\Gamma(v_0)_2$ and $\Gamma(v_0)_3$ should be done by deleting the $c(v_0)$ domain from their $\gamma(v_0)$ strand (we use $\gamma'(v_0)$ to denote the remainder of $\gamma(v_0)$ after deleting its $c(v_0)$ domain). Correspondingly, the input single-stranded $Input(v_0)$ is replaced by three short single strands $Input(v_0)_1$, $Input(v_0)_2$, and $Input(v_0)_3$ (see Supplementary Fig. S10 e), where $Input(v_0)_i$ is the complementary $\beta_i^*(v_0)$ of $\beta_i(v_0)$, for $i = 1, 2, 3$. Because $\gamma'(v_0)$ and $D(v_0)$ do not produce fluorescence signal (see the previous experiments for the MIS problem, as shown in Fig. 4 b), the strand displacement reaction for computation gate (Eq. (1)) can be replaced by the following three reactions.



Application to SAT. In the previous sections, we demonstrated that DCMSubset can be used to solve subset problems in graph theory. To further extend the scope of its applicability, we attempted to solve other problems by DCMSubset. A DCMSubset-based approach for the SAT problem is described as follows.

Problem 5 The SAT problem aims to determine whether there is a truth assignment of 0 ('false') or 1 ('true') to a set of boolean variables $\{x_1, \dots, x_n\}$ that makes a conjunctive normal form (CNF) $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$ true, where $C_i = \ell_{i1} \vee \ell_{i2} \vee \dots \vee \ell_{ik_i}$ for $i = 1, 2, \dots, m$ and ℓ_{ij} ($j = 1, 2, \dots, k_i$) is called a literal, which is either x or the negative $\neg x$ of x for some $x \in \{x_1, \dots, x_n\}$.

It is clear that F has true value if and only if every clause C_i has true value. Our target is to design a DNA circuit based on the method of DCMSubset to determine the truth value of each clause C_i for a given truth assignment to the boolean variables.

Unlike in the models for graph problems, for each $i \in \{1, 2, \dots, n\}$, we used single strands $\gamma^+(x_i)$ and $\gamma^-(x_i)$ (see Supplementary Fig. S6 a) to present x_i and $\neg x_i$, respectively. Here, $\gamma^+(x_i)$ and $\gamma^-(x_i)$ were designed as follows: In the direction from 5' to 3', $\gamma^+(x_i)$ consists of a toehold domain $a(x_i)$ and two branch migration domains $b(x_i)$ and $c(x_i)$; $\gamma^-(x_i)$ consists of the complementary domains $a(x_i)^*$, $b(x_i)^*$, and $c(x_i)^*$

of $a(x_i)$, $b(x_i)$, and $c(x_i)$. Observe that although the bases between $\gamma^+(x_i)$ and $\gamma^-(x_i)$ are complementary along the 5' to 3' direction, they are not hybridized into a double strand owing to the restriction of their directions.

The computation gate $\Gamma(C_i)$ was designed for each clause C_i , $i = 1, 2, \dots, m$, obtained by binding all $\gamma(x)$ for $x \in C_i$ (here, $\gamma(x) = \gamma^-(x)$ if the corresponding literal is the negative form of x ; otherwise, $\gamma(x) = \gamma^+(x)$) to a single-stranded $\beta(C_i)$, via the toehold domain $a(x)$ (or $a(x)^*$) and branch migration domain $b(x)$ (or $b(x)^*$). Note that $\beta(C_i)$ was designed different from the previous design; we reserved a toehold domain t for every $x \in C_i$; see Supplementary Fig. S6 c.

Now, for each truth assignment $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$, where $\theta_i \in \{0, 1\}$ represents the truth value of the variable x_i , we designed n input strands $Input(x_i)$ (see Supplementary Fig. S6 b), $i = 1, 2, \dots, n$ as follows: When $\theta_i = 0$, $Input(x_i)$ is denoted by $Input^-(x_i)$, which (in the direction from 5' to 3') consists of the toehold domain t^* and the complementary domains $a(x_i)^*$ and $b(x_i)^*$ of $a(x_i)$ and $b(x_i)$, respectively; when $\theta_i = 1$, $Input(x_i)$ is denoted by $Input^+(x_i)$, which (in the direction from 5' to 3') consists of the toehold domain t^* and two branch migration domains $a(x_i)$ and $b(x_i)$. For the same reason as that for $\gamma^+(x_i)$ and $\gamma^-(x_i)$, $Input^+(x_i)$ and $Input^-(x_i)$ cannot hybridize into a double-stranded complex.

Each computation gate $\Gamma(C_i)$ receives k_i input signal strands $Input(x_{ij})$, $j = 1, 2, \dots, k_i$ as the invader to displace the incumbent strands $\gamma(x_{ij})$, where $C_i = \ell_{i1} \vee \ell_{i2} \vee \dots \vee \ell_{ik_i}$. If and only if $Input(x_{ij})$ and $\gamma(x_{ij})$ have the same symbol (i.e., either $Input^+(x_{ij})$ and $\gamma^+(x_{ij})$, or $Input^-(x_{ij})$ and $\gamma^-(x_{ij})$), the incumbent strands $\gamma(x_{ij})$ can be replaced by the strand displacement reaction of this group of inputs and $\Gamma(C_i)$.

The detection gate $D(\ell_{ij})$ was designed for each literal in C_i , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, k_i$, which is obtained by binding the target strand (a complementary strand $\gamma(\ell_{ij})^*$ of $\gamma(\ell_{ij})$) to a single-stranded output, via the branch migration domains $b(x)$ (or $b(x)^*$) and $c(x)$ (or $c(x)^*$). When the invader $\gamma(\ell_{ij})$ exists, the strand displacement reaction of $\gamma(\ell_{ij})$ and $D(\ell_{ij})$ can occur and produce an output single. Supplementary Fig. S6 d illustrates the work flow diagram of DCMSubset on SAT. In Supplementary Fig. S6 c, we give a molecular implementation of the DCMSubset circuit for the SAT problem.

For each truth assignment θ , we carried out i ($i = 1, 2, \dots, m$) groups of experiments to determine the truth value of each clause C_i . T_i denotes the i th group of experiments. Re-grading each group of experiments, we added $\Gamma(C_i)$ (10nM), $D(\ell_{ij})$ for $j = 1, 2, \dots, k_i$ (each 10nM), and inputs corresponding to the truth assignment (each $Input(x_i)$ 10nM and total $(10 * n)nM$) to T_i . Then, we detected the content change of output signal by DSD simulation. According to the above analysis, the truth value of C_i is true if and only if the output content increases in T_i . Therefore, F is true under θ if and only if the output content increases in all these m groups of experiments. See the supplementary case analysis for an example of solving the SAT problem.

Based on our results, we posit that DCMSubset can also

be used to design DNA circuits for other classes of NP-hard problems, such as the Hamiltonian problem, which we will attempt to realize in our future work.

Methods

Materials and apparatuses. The DNA oligonucleotides (oligos) used in this study were purchased from Sangon Biotech (Shanghai). All DNA oligos were purified by Sangon using HPLC. Agarose, 10×TE Buffer, 50×TAE Buffer, 4S Green Plus Nucleic Acid Stain, Sterilized ddH₂O, and DNA Marker A (25-500bp) were purchased from Sangon Biotech. The DNA loading buffer (6×) was purchased from biosharp (Guangzhou). The concentration of DNA oligonucleotides was measured using a NanoPhotometer[®] N120 (Implen, Munich, Germany). All samples were annealed in a polymerase chain reaction thermal cycler (Thermo Fisher Scientific Inc., USA). The PCR analyses were carried out in QuantStudio 3, using the QuantStudio Design & Analysis Software v.1.5.1 (Applied Biosystems, MA, US). Agarose gel electrophoresis was carried out in BG-subMINI with BG-Power 600 (BayGene, Beijing). The DNA image on the gel was taken on AlphaImager HP (Protein Simple, San Jose, CA) gel documentation system.

Individual unlabeled DNA oligos were dissolved in 1×TE buffer (nuclease free, pH 7.8~8.2) containing 12.5 mM Mg²⁺ to a final concentration of 100 μM according to the information provided by the supplier, and stored at 4 °C. Oligos labeled with dyes or quenchers were dissolved in Sterilized ddH₂O to a final concentration of 100 μM, according to the information provided by the supplier, and stored in light-proof tubes at 4 °C.

DNA sequences. All the DNA sequences used in our study are listed in Supplementary Tables S1 and S2. The DNA sequences were designed to meet several requirements: (a) To balance the reaction rate, the GC content of all DNA strands was maintained at 50% ~ 70%; (b) We ensured the stability of the formed molecule; (c) To reduce the leakage of logic gates in the DCMSubset model, we avoided duplication of three consecutive bases at the same position in all toehold domains. Initially, the original sequences were obtained by using Nupack, and were then modified by hand.

Molecular Assembly. All sequences that were assembled had to be checked to ensure that no secondary structures affecting strand displacement were generated. We found by NUPACK check that the generation rate of all logic gate secondary structures was not higher than 0.2% at 37°C. Here, we used the NUPACK to obtain the minimum free energy structure (listed in Supplementary Table S3), where the relevant parameters required for simulation are shown in Supplementary Table S4. All $\Gamma(x)$ complexes and $\Gamma'(x)$ complexes (listed in Supplementary Table S1) were assembled by mixing the corresponding single strands with equal molar concentrations and equal volumes. Moreover, all $D(x)$ complexes (listed in Supplementary Table S1) were assembled by mixing the corresponding single strands using the concentration ratio $\gamma(x)^*$:

output = 1.1 : 1. Depending on the buffer for dissolving $\gamma(x)^*$ and *output* (which was sterilized ddH₂O), to ensure the same ion concentration of $\Gamma(x)$ and $D(x)$, we added an additional 3× TAE containing 37.5 mM Mg²⁺ to neutralize the ion concentration when assembling $D(x)$, in addition to mixing $\gamma(x)^*$ and *output*. Further, $\gamma(x)^*$, *output*, and 3× TAE/Mg²⁺ were mixed with equal volume. We obtained the concentrations of the diluted DNA single strands using the instrument. We took two approaches to ensure the accuracy of the measured concentrations: (1) the solution was fully shaken and centrifuged before each measurement; and (2) three measurements were taken and averaged as the final concentration used.

Considering that length and number of single strands were different for the assembled computation and detection gates, we use two different annealing procedures to assemble logic gates. The computation gates $\Gamma(x)$ and $\Gamma'(x)$ were first held at 95°C for 3 minutes, then cooled to 60°C at a rate of 1°C/min, and then cooled to 4°C at a rate of 0.33°C/min. Finally, the computation gates were stored at 4 °C. The detection gates were first held at 95°C for 3 minutes, then cooled to 35°C at a rate of 0.6°C/min, and then cooled to 4°C at a rate of 1°C/min. Finally, the detection gates were stored in the dark at 4°C. These steps were performed on a PCR thermal cycler.

Agarose gel electrophoresis. The DNA solutions were analyzed in a 5% Agarose gel in a 1× TAE buffer after running for 25 min at a constant power of 110 V using BG-subMINI and BG-Power 600, and imaged with the gel documentation system. 1× TAE was obtained by using ultrapure water to dilute 50× TAE. To prevent DNA samples from running out of the gel during electrophoresis, we used a DNA loading buffer (6×) (2μl) to color the DNA samples (8μl) before performing electrophoresis. Also, we used DNA Marker A (25-500bp) (7μl) as a reference for the DNA samples (10μl).

The 5% agarose gel was prepared as follows: First, agarose (2.5 g powder) was dissolved in a 1×TAE buffer (50 ml); then, it was heated until the powder was completely dissolved, and 4S Green Plus Nucleic Acid Stain (10 μl) was added to dissolve it fully; finally, it was poured into a membrane tool with a comb and cooled into a solid before being prepared for use.

Simulation and fluorescence kinetics experiments. Simulation for dynamic analysis was completed by Visual DSD. The simulation duration was set to 600 s.

All fluorescence kinetic detection was performed using the QuantStudio 3 equipped with a 96-well fluorescence plate reader. The fluorescence detection procedure was as follows: First, in the Hold Stage, the temperature was decreased to 4 °C by a rate of 1.6 °C per second, and then the sample was quickly put into QuantStudio 3; second, in the PCR Stage, the temperature was increased to 23 °C by a rate of 3 °C per second; finally, the fluorescence intensity of the sample was measured every 3 min at 23 °C.

References

1. Xu, J., Qiang, X., Zhang, K., Zhang, C. & Yang, J. A DNA computing model for the graph vertex coloring problem based on sa probe graph. *Engineering* **4**, 61–77 (2018).
2. Tang, Z. *et al.* Solving 0–1 integer programming problem based on DNA strand displacement reaction network. *ACS Synth. Biol.* **10**, 2318–2330 (2021).
3. Benenson, Y. DNA computes a square root. *Nat. Nanotechnol* **6**, 465–467 (2011).
4. Xu, J., Chen, C. & Shi, X. Graph computation using algorithmic self-assembly of DNA molecules. *ACS Synth. Biol.* **11**, 2456–2463 (2022).
5. Qian, L., Winfree, E. & Bruck, J. Neural network computation with DNA strand displacement cascades. *Nature* **475**, 368–372 (2011).
6. Qian, L. & Winfree, E. Scaling up digital circuit computation with DNA strand displacement cascades. *Science* **332**, 1196–1201 (2011).
7. Jung, J. K., Archuleta, C. M., Alam, K. K. & Lucks, J. B. Programming cell-free biosensors with DNA strand displacement circuits. *Nat. chem biol* **18**, 385–393 (2022).
8. Jung, J. K. *et al.* Cell-free biosensors for rapid detection of water contaminants. *Nat. biotechnol* **38**, 1451–1459 (2020).
9. Adleman, L. M. Molecular computation of solutions to combinatorial problems. *Science* **266**, 1021–1024 (1994).
10. Reif, J. H. Parallel molecular computation. In *Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures*, 213–223 (1995).
11. Lo, Y.-M., Yiu, K. & Wong, S. On the potential of molecular computing. *Science* **268**, 481–482 (1995).
12. Lipton, R. J. DNA solution of hard computational problems. *Science* **268**, 542–545 (1995).
13. Roweis, S. *et al.* A sticker-based model for DNA computation. *J. Comput. Biol.* **5**, 615–629 (1998).
14. Zimmermann, K.-H. Efficient dna sticker algorithms for NP-complete graph problems. *Comput. Phys. Commun.* **144**, 297–309 (2002).
15. Braich, R. S., Chelyapov, N., Johnson, C., Rothmund, P. W. & Adleman, L. Solution of a 20-variable 3-SAT problem on a dna computer. *Science* **296**, 499–502 (2002).
16. Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C. & Neumann, J. L. A DNA-fuelled molecular machine made of dna. *Nature* **406**, 605–608 (2000).
17. Liu, C. *et al.* Cross-inhibitor: a time-sensitive molecular circuit based on DNA strand displacement. *Nucleic Acids Res.* **48**, 10691–10701 (2020).

18. Zhang, D. Y. & Winfree, E. Control of dna strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.* **131**, 17303–17314 (2009).
19. Machinek, R. R., Ouldrige, T. E., Haley, N. E., Bath, J. & Turberfield, A. J. Programmable energy landscapes for kinetic control of DNA strand displacement. *Nat. Commun.* **5**, 1–9 (2014).
20. Wang, F. *et al.* Implementing digital computing with dna-based switching circuits. *Nat. Commun.* **11**, 1–8 (2020).
21. Zhu, E., Luo, X., Liu, C. & Chen, C. An operational DNA strand displacement encryption approach. *Nanomaterials* **12**, 877 (2022).
22. Tang, Z., Yin, Z., Yang, J., Sun, X. & Cui, J. The circular DNA model of 0–1 programming problem based on dna strand displacement. In *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 173–177 (IEEE, 2018).
23. Yang, J., Pang, X., Tang, Z., Yang, X. & Liu, C. DNA strand displacement computing model for the sat problem. In *Journal of Physics: Conference Series*, vol. 2026, 012040 (IOP Publishing, 2021).
24. Zhang, X., Zhang, W., Zhao, T., Wang, Y. & Cui, G. Design of logic circuits based on combinatorial displacement of DNA strands. *J. Comput. Theor. Nanosci.* **12**, 1161–1164 (2015).
25. Yang, C. *et al.* A versatile DNA-supramolecule logic platform for multifunctional information processing. *NPG Asia Mater.* **10**, 497–508 (2018).
26. Woods, D. *et al.* Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature* **567**, 366–372 (2019).
27. Currin, A. *et al.* Computing exponentially faster: implementing a non-deterministic universal turing machine using DNA. *J. R Soc. Interface* **14**, 20160990 (2017).
28. Lee, W., Yu, M., Lim, D., Kang, T. & Song, Y. Programmable DNA-based boolean logic microfluidic processing unit. *ACS Nano* **15**, 11644–11654 (2021).
29. Cook, M. *et al.* Universality in elementary cellular automata. *Complex Syst.* **15**, 1–40 (2004).
30. Margulies, D., Melman, G. & Shanzer, A. A molecular full-adder and full-subtractor, an additional step toward a molecular calculator. *J. Am. Chem. Soc.* **128**, 4865–4871 (2006).
31. Zhou, C., Geng, H., Wang, P. & Guo, C. Programmable DNA nanoindicator-based platform for large-scale square root logic biocomputing. *Small* **15**, 1903489 (2019).
32. Xie, T. *et al.* DNA circuits compatible encoder and demultiplexer based on a single biomolecular platform with dna strands as outputs. *Nucleic Acids Res.* **50**, 8431–8440 (2022).
33. Shah, S. *et al.* Using strand displacing polymerase to program chemical reaction networks. *J. Am. Chem. Soc.* **142**, 9587–9593 (2020).
34. Reid, M. S., Le, X. C. & Zhang, H. Exponential isothermal amplification of nucleic acids and assays for proteins, cells, small molecules, and enzyme activities: an example. *Angew. Chem. Int. Ed.* **57**, 11856–11866 (2018).
35. Kang, D. *et al.* DNA biomolecular-electronic encoder and decoder devices constructed by multiplex biosensors. *NPG Asia Mater.* **4**, e1–e1 (2012).
36. de Luis, B. *et al.* A 1-to-2 demultiplexer hybrid nanocarrier for cargo delivery and activation. *Chem. Commun.* **56**, 9974–9977 (2020).
37. Thomas, P. *et al.* 1-to-4 analog demultiplexer with up to 128 gs/s for interleaving of bandwidth-limited digitizers in wireline and optical receivers. *IEEE. J. Solid-State Circuits* **56**, 2611–2623 (2021).
38. Reed, B. Paths, stars and the number three. *Comb. Probab. Comput.* **5**, 277–295 (1996).
39. Lovász, L. On the shannon capacity of a graph. *IEEE. Trans. Inf. Theory* **25**, 1–7 (1979).
40. Brooks, R. L. On colouring the nodes of a network. *Math. Proc. Camb. Philos. Soc.* **37**, 194–197 (1941).
41. Gallai, T. Über extreme punkt- und kantenmengen. *Annales Univ. Sci. Budapestinensis de Rolando Eötvös Nominatae, Sect. Math.* **2**, 133–138 (1959).

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant 61872101; in part by Natural Science Foundation of Guangdong Province of China under Grant 2021A1515011940.

Author contributions

Conceptualization, E.Z. and X.L.; Investigation, E.Z., X.L. and C.L.; Methodology, E.Z., X.L. and X.S.; Supervision, J.X. and X.S.; Writing-original draft, E.Z. and X.L.; Writing-review and editing, E.Z. and C.L. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no conflict of interest.

Supplementary information Supplementary information is available for this paper at [here](#).