

基于粘贴和删除系统的图着色问题分析

王淑栋

(山东科技大学信息科学与工程学院 山东 青岛 266510)
(北京大学信息科学技术学院软件所 北京 100871)

摘 要 图着色问题是图与组合优化中的一个 NP-完全问题. 现有算法在求解图着色问题时, 计算复杂性随着待解决问题规模的增大呈指数增长. 粘贴系统和删除系统是分别基于粘贴运算和删除运算的两种语言生成器. 文中将图着色问题和图的坏边数结合起来, 将图着色问题转化成搜索最长序列的问题, 然后利用粘贴系统和删除系统的并行性, 得到了图的色数及其所有色类. 与已有求解图着色问题的 DNA 算法相比, 新的算法具有较低的复杂性.

关键词 DNA 计算; 图着色; 粘贴和删除系统; 坏边
中图法分类号 TP301

Analysis for Graph Coloring Problem Based on Sticker and Deletion Systems

WANG Shu-Dong

(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266510)
(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

Abstract Graph coloring problem is a NP-complete one of graph and combinatorial optimization. The computational time increases exponentially with the size of the researched problem solved using the usual methods. Thus it is impossible to settle it efficiently. Sticker system and deletion system are the language generated mechanisms based on sticking and deleting operations respectively. In this paper, graph coloring problem was combined with the number of bad edges and transformed into a problem of searching for the longest sequence. On the basis, the chromatic number and all the color classes of graph were obtained using the parallelism of two formal models: sticker system and deletion system. Compared with the existing DNA algorithm for graph coloring problem, the proposed algorithm is of lower complexity.

Keywords DNA computing; graph coloring; sticker and deletion systems; bad edge

1 Introduction

In 1998, Kari^[1] introduced the definition of the sticker system in the form of simple regular sticker system as a formal model for self-assembly computation. The sticker system most clearly represents and formalizes a computation process of the self-assembly mechanisms. It formalizes the sticker operation, which can build double-stranded

sequences starting from “axioms” using single-stranded sequences or sequences with sticky ends at one or both their ends called “DNA dominoes”, by annealing to stick to each other. Afterwards, Păun et al.^[2] introduced the sticker system prolonging the sequences in both directions. In 2001, Sakakibara^[3] proposed a variant of sticker system, which uses molecules with complex structures. The type of sticker system may be seen as one of

simple regular sticker systems in the original definition. In 1996, Kari and Thierrin^[4] introduced the concept of insertion-deletion system, following the systematic study of insertion-deletion operations in [5]. Here, only deletion operation (without using insertion operation) is considered.

Mathematically, a one-one assignment of colors to the vertices of graph G such that any two adjacent vertices are assigned different colors is called a coloring of G . A coloring in which k colors are used is a k -coloring. A graph G is k -colorable if there exists a k -coloring of G . The minimum number k such that graph G admits a k -coloring is called the vertex chromatic number, or simply the chromatic number of G , denoted by $\chi(G)$. In a given coloring of G , a set consisting of all those vertices assigned the same color is called a color class. The graph coloring problem is: Given a graph G with n vertices and m edges, how can the graph be colored by the minimum chromatic number? What is $\chi(G)$? And what are the color-classes? Furthermore, whether it is uniquely $\chi(G)$ -colorable? As is well-known, the graph coloring problem is an NP-complete problem. In recent years, much work has been done to search better

ways based on biomolecular computing for graph coloring problem^[6-11]. This paper introduces a novel approach based on sticker and deletion systems for graph coloring problem and makes comparison with some existing algorithms in the fifth section of this paper.

2 Preliminaries and Lemmas

An alphabet is a finite nonempty set of abstract symbols. For an alphabet V , denote by V^* the set of all strings of symbols in V . The empty string is denoted by λ . Each subset of V^* is called a language over V .

Consider an alphabet V and a symmetric relation $\rho \subseteq V \times V$ (of complementarity). Denote

$$\begin{aligned} \begin{bmatrix} V \\ V \end{bmatrix}_\rho &= \left\{ \begin{bmatrix} a \\ b \end{bmatrix} \mid a, b \in V, (a, b) \in \rho \right\}, WK_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^*, \\ LR_\rho(V) &= \left(\left(\begin{smallmatrix} \lambda \\ V^* \end{smallmatrix} \right) \cup \left(\begin{smallmatrix} V^* \\ \lambda \end{smallmatrix} \right) \right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^+ \left(\left(\begin{smallmatrix} \lambda \\ V^* \end{smallmatrix} \right) \cup \left(\begin{smallmatrix} V^* \\ \lambda \end{smallmatrix} \right) \right) \end{aligned}$$

(see Fig. 1).

$$L_\rho(V) = \left(\left(\begin{smallmatrix} \lambda \\ V^* \end{smallmatrix} \right) \cup \left(\begin{smallmatrix} V^* \\ \lambda \end{smallmatrix} \right) \right) \begin{bmatrix} V \\ V \end{bmatrix}_\rho^* \quad (\text{see Fig. 2(a), (b)}).$$

$$R_\rho(V) = \begin{bmatrix} V \\ V \end{bmatrix}_\rho^* \left(\left(\begin{smallmatrix} \lambda \\ V^* \end{smallmatrix} \right) \cup \left(\begin{smallmatrix} V^* \\ \lambda \end{smallmatrix} \right) \right) \quad (\text{see Fig. 2(c), (d)}).$$

$$W_\rho(V) = L_\rho(V) \cup R_\rho(V) \cup LR_\rho(V).$$

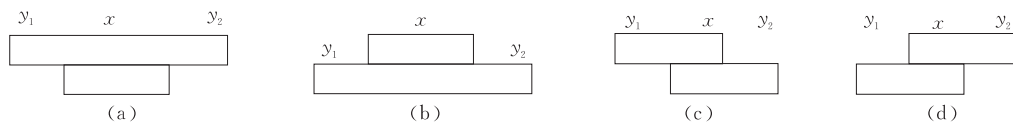


Fig. 1 Incomplete double-stranded molecules with two sticky ends, where x is a double-stranded sequence and y_1, y_2 are single-stranded sequences.

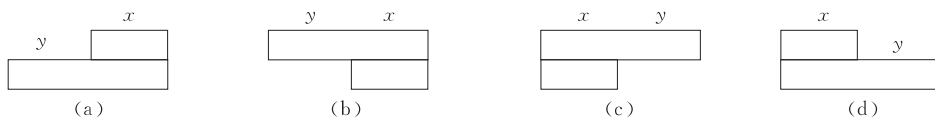


Fig. 2 Incomplete double-stranded molecules with one sticky end, where x is a double-stranded sequence and y is a single-stranded sequence.

A sticker system is a construct $\gamma = (V, \rho, A, D)$, where V is an alphabet, $\rho \subseteq V \times V$ is a symmetric relation, A is a finite subset of $W_\rho(V)$, and D is a finite subset of $W_\rho(V) \times W_\rho(V)$. The elements of A are called axioms. A sequence $x_1 \Rightarrow x_2 \Rightarrow \dots \Rightarrow x_k$, with $x_1 \in A$ is called a computation in γ . A computation $\sigma: x_1 \Rightarrow^* x_k$ is complete when $x_k \in WK_\rho(V)$. The set of all molecules produced at the end of complete computation in γ is denoted by $LM_n(\gamma)$. Namely $LM_n(\gamma) = \{\omega \in WK_\rho(V) \mid x \Rightarrow^* \omega, x \in A\}$. The language generated by sticker system γ is defined by

$$L_n(\gamma) = \left\{ \omega \in V^* \mid \begin{bmatrix} \omega \\ \omega' \end{bmatrix} \in LM_n(\gamma), \omega' \in V^* \right\}.$$

A deletion system is a construct $\gamma = (V, T, A, R)$, where V is an alphabet, $T \subseteq V$, A is a finite language over V , and R is a finite set of triples of the form $(u, \alpha/\lambda, v)$, $u, v \in V^*$, $\alpha \in V^+$. The elements of T are terminal symbols, those of A are axioms, the triples in R are deletion rules. The meaning of $(u, \alpha/\lambda, v)$ is that α can be deleted from the context (u, v) . For $x, y \in V^*$, we can write $x \Rightarrow y$ if y can be obtained from x by using a deletion rule. The language generated by deletion system γ is defined by $L(\gamma) = \{\omega \in T^* \mid x \Rightarrow^* \omega, x \in A\}$.

Let $G = (V, E)$ be a graph with vertex-set $V = \{v_1, v_2, \dots, v_n\}$. The set of all the permutations on the set V is denoted by S_n . Each element s of S_n is

called an ordering of S_n . Let $s = v_1 v_2 \cdots v_n$ and $i, j \in \{1, 2, \dots, n\}, i < j$. Denote by $I[s, i, j]$ the interval determined by the two subscripts i and j in s , that is $I[s, i, j] = v_{i+1} v_{i+2} \cdots v_{j-1}$. Let $e \in E$ be an edge of G and $s \in S_n$ an ordering. Call e a bad edge with respect to s if $e = v_i v_j, i < j$ and $I[s, i, j]$ is a stable set (independent set) in G . For each ordering $s \in S_n$, Denote by $b(s)$ the number of all bad edges in G with respect to s .

Based on the above definitions, Croitoru^[12] gave a theorem which showed that determining an optimal coloring for a graph is equivalent to looking for an ordering of its vertices with a minimum number of bad edges.

Lemma 1.^[12] There exists a sequence $s = v_{i_1} v_{i_2} \cdots v_{i_n} \in S_n$ such that $b(s) = |\{v_{i_j} v_{i_{j+1}} \in E | j = 1, 2, \dots, n-1\}| = \min_{s \in S_n} b(s)$.

Lemma 2.^[12] Let G be a graph. Then $\chi(G) = 1 + \min_{s \in S_n} b(s)$.

The undefined terms and notations in this paper can refer to [12] and [13].

3 Theory Designs of Graph Coloring

3.1 Construction of Sticker System

Let $G = (V, E)$ be a graph with n vertices and m edges, where $V = \{v_1, v_2, \dots, v_n\}$ and $E = \{e_1, e_2, \dots, e_m\}$ are the vertex-set and edge-set of G respectively.

Theorem 1. There exists a sticker system generating the language from which we can obtain all the permutations over vertex-set V of G .

Proof. We construct the sticker system γ_1 as follows. Let $\gamma_1 = (V', \rho, A, D)$, where $V' = V \cup \{M\}$, M is a symbol, $\rho = \{(v_i, \bar{v}_i) | v_i \in V\} \cup \{(M, \bar{M})\}$, $A = \left\{ \begin{bmatrix} M \\ \bar{M} \end{bmatrix} \binom{v_i}{\lambda}, i = 1, 2, \dots, n \right\}$, $D = \left\{ \begin{pmatrix} M v_i \\ \lambda \end{pmatrix}, \begin{pmatrix} \lambda \\ v_i M \end{pmatrix}, i = 1, 2, \dots, n \right\} \cup \left\{ \begin{pmatrix} M \\ \lambda \end{pmatrix} \right\}$.

Obviously, the sequences in A can be prolonged only to the right by using the columns of the form $\begin{pmatrix} \lambda \\ v_i M \end{pmatrix}, i = 1, 2, \dots, n$, afterwards following by adding the columns $\begin{pmatrix} M \\ \lambda \end{pmatrix}$ or $\begin{pmatrix} M v_i \\ \lambda \end{pmatrix}, i = 1, 2, \dots, n$ to the right. The above two steps are proceeded alternately. But once $\begin{pmatrix} M \\ \lambda \end{pmatrix}$ is applied, the computation must be halted. Finally, we obtain all the double-stranded sequences which are constituted by $\begin{pmatrix} M \\ \bar{M} \end{pmatrix}$ and $\begin{pmatrix} v_i \\ \bar{v}_i \end{pmatrix}$ sandwiched sequentially, $i =$

$1, 2, \dots, n$. Namely, sticker system γ_1 generates the sequences of the form $M v_{i_1} M v_{i_2} M \cdots M v_{i_j} M$, where $v_{i_1}, v_{i_2}, \dots, v_{i_j} \in \{v_1, v_2, \dots, v_n\}$. It is easy to see that the language $L_n(\gamma_1)$ generated by sticker system γ_1 satisfies the above requirement. \square

Subsequently, we use the separating technique of magnetic beads and get all the sequences which contain $v_i, i = 1, 2, \dots, n$ as substrings. Following by an operation of measuring the length of sequences, we gain all the sequences of length $2n+1$. For the sake of descriptive convenience, we call the obtained sequences the standard sequences.

3.2 Construction and Analysis of Deletion System

Theorem 2. There exists a deletion system generating the language from which we can obtain all the orderings including the least bad edges.

Proof. Construct the deletion system γ_2 as follows. Let $\gamma_2 = (V', T, A, R)$, where $V' = V \cup \{M\}$, $T = V'$, A is the set of all the standard sequences, and $R = \{(v_i, M/\lambda, v_j) | v_i v_j \in E\}$. According to the construction of deletion system above, we may draw the conclusion that, in any standard sequences, if the character v_i before M is adjacent to the character v_j after the same M in graph G , that is $v_i v_j \in E$, then deletion system γ_2 will delete the M between v_i and v_j . Again by the definition of bad edge, $v_i v_j$ is a bad edge. Let the length of the longest sequences be l and s be a sequence with length l in $L(\gamma_2)$. Then by Lemma 1, the number of deleting M in s equals the minimum number of bad edges. Consequently, the longest sequences in $L(\gamma_2)$ includes the least bad edges. Furthermore, the minimum bad edge number is $2n+1-l$. \square

For the sake of further descriptive convenience, we call the longest sequences in $L(\gamma_2)$ feasible sequences.

Again by Lemma 2 and Theorem 2, we can immediately obtain the following theorem.

Theorem 3. Let the length of the longest sequences be l in $L(\gamma_2)$. Then the chromatic number of graph G $\chi(G) = 2n+2-l$.

4 DNA Algorithm Design of Graph Coloring

4.1 DNA Algorithm of Graph Coloring

Our algorithm can be described as follows:

Step 1. Generate all the possible vertex-sequences;

Step 2. Keep only those sequences which contain every vertex exactly once;

Step 3. Find out all the feasible sequences.

4.2 Implementation of the DNA Algorithm

A tube is a multiset of words (finite strings)

over $\{A, C, G, T\}$. Intuitively, a tube is a collection of DNA strands. In our algorithm, the following operations are needed.

Step1. Separate. Given a tube N and a word w over the alphabet $\{A, C, G, T\}$, produce two tubes $+(N, w)$ and $-(N, w)$, where $+(N, w)$ consists of all strands in N which contain w as a substring and similarly, $-(N, w)$ consists of all strands in N which do not contain w as a substring.

Step2. Length-separate. Given a tube N and an integer n , produce the tube (N, n) consisting of all strands of length n in N .

Step3. To-Single-Stranded. Denature each dsDNA in tube and remove one ssDNA.

The above three operations can be implemented using the existing biomolecular technology.

To implement Step 1 of the algorithm, first we need encode $n+1$ oligonucleotides v_1, v_2, \dots, v_n, M and synthesize $3n$ DNA sequences $Mv_i, \overline{v_iM}$ and $\left[\begin{smallmatrix} M \\ \overline{M} \end{smallmatrix} \right] \left(\begin{smallmatrix} v_i \\ \lambda \end{smallmatrix} \right), i=1, 2, \dots, n$, where the over bar “ $-$ ” represents the complementary sequence. Then we put plenty of the above synthesized $3n$ DNA sequences into tube N and produce the double-stranded sequences which are constituted by M and $v_i, i=1, 2, \dots, n$ sandwiched sequentially using those operations of sticker system γ_1 of 3.1. Namely, $\left[\begin{smallmatrix} Mv_iMv_{i_2}M \cdots Mv_{i_j}M \\ \overline{Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M} \end{smallmatrix} \right],$ where $v_{i_1}, v_{i_2}, \dots, v_{i_j} \in \{v_1, v_2, \dots, v_n\}$.

To implement Step 2 of the algorithm, first we must use the operation To-Single-Stranded to make the above double-stranded sequences $\left[\begin{smallmatrix} Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M \\ \overline{Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M} \end{smallmatrix} \right]$ to single-stranded ones $Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M$. Then using n “Separate” operations $+(N, v_i)$ and $-(N, v_i), i=1, 2, \dots, n$, we get all the sequences which contain $v_i, i=1, 2, \dots, n$ as substrings. Following by a “Length-separate” operation $(N, 2n+1)$, we gain all the sequences of length $2n+1$ and including $v_i, i=1, 2, \dots, n$ as substrings.

To implement Step 3 of the algorithm, first we must use the operations in the deletion system γ_2 of 3.2. Then we obtain all the feasible sequences using polyacrylamide gel electrophoresis.

If you also ask further, what are the color-classes? We need to proceed with the operations of reading out the feasible sequences. If the vertices and M arrange alternately, then these vertices be-

long to the same color-class; If the vertex joins another one, then the two vertices do not belong to the same color-class. For example, the feasible sequence is $Mv_{i_1}v_{i_2}v_{i_3}Mv_{i_4}M$, then the color-classes are $\{v_{i_1}\}, \{v_{i_2}\}, \{v_{i_3}, v_{i_4}\}$. For any consecutive $v_i v_j$, vertex v_i belongs to the color-class to which the vertex does with symbol M before v_i ; Vertex v_j belongs to the color-class to which the vertex does with symbol M after v_j ; The vertex self belongs to one color-class if symbol M is neither before nor after the one.

Now we give an example (Fig.3) to show the whole process of the above implementation. Firstly, we encode 6 vertices v_1, v_2, \dots, v_6 and one symbol M as Table 1, where vertex v_i and M correspond to the sequences of length 14 bases and 10 ones respectively, $i=1, 2, \dots, 6$, and synthesize 18 DNA sequences as Table 2. Then we put plenty of the above synthesized 18 DNA sequences into tube N and obtain doubled-sequences $\left[\begin{smallmatrix} Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M \\ \overline{Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M} \end{smallmatrix} \right], v_{i_1}, v_{i_2}, \dots, v_{i_j} \in \{v_1, v_2, \dots, v_6\}$ by using sticking operations of sticker system γ_1 of 3.1. Subsequently, we proceed with the operation of To-Single-Stranded to the above doubled-sequences and get single-sequence $Mv_{i_1}Mv_{i_2}M \cdots Mv_{i_j}M$. After six “Separate” operations $+(N, v_i), -(N, v_i), i=1, 2, \dots, 6$ and a “Length-separate” operation $(N, 154)$, we gain $6! (=720)$ sequences of length 154 bases and including v_i sequence as substrings, $i=1, 2, \dots, 6$, where 154 is the length of sequences containing 6 vertices and 7 M sequences, i. e. $154 = 10 \times 7 + 14 \times 6$. Then we apply the deleting operations in the deletion system γ_2 of 3.2 to the above 720 sequences. For instance, for sequence $Mv_1Mv_2Mv_3Mv_4Mv_5Mv_6M$, since $v_1 v_2, v_2 v_3, v_3 v_4, v_4 v_5, v_5 v_6$ are the edges of Fig. 3, according to the deleting rules of deletion system γ_2 , the five M s are deleted between v_1 and v_2, v_2 and v_3, v_3 and v_4, v_4 and v_5, v_5 and v_6 , and obtain the sequence $Mv_1v_2v_3v_4v_5v_6M$ of length 104 bases. For another sequence $Mv_1Mv_4Mv_2Mv_5Mv_3Mv_6M$, since $v_1 v_4, v_2 v_5, v_3 v_6$ are not and $v_4 v_2, v_5 v_3$ are the edges of Fig. 3, the two M s are deleted between v_4 and v_2, v_5 and v_3 , and obtain the sequence $Mv_1Mv_4v_2Mv_5v_3Mv_6M$ of length 134 bases. Afterwards, we pick out all the longest sequences (i. e. all the feasible sequences) using polyacrylamide gel electrophoresis. In this example, the length of the feasible sequences is 134 bases. By Theorem 3, the minimum bad edge number is $\frac{154-134}{10} = 2$ and by Lemma 2,

$\chi(G) = 3$. The color-classes corresponding to sequence $Mv_1Mv_4v_2Mv_5v_3Mv_6M$ are $\{v_1, v_4\}$, $\{v_2, v_5\}$ and $\{v_3, v_6\}$.

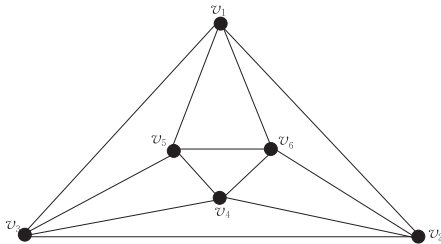


Fig. 3 Octahedron—The graph of coloring problem

Table 1 The Encoded DNA Sequences Corresponding to Vertices and M	
Encoded characters	DNA sequences (from 5' to 3')
v_1	ACCTGTACTGATCG
v_2	ATGCTGAATGGCCT
v_3	TCGAACCTATCACG
v_4	TAAGGTCGGCTGAT
v_5	ATGCTAGCTACCGT
v_6	GTCACGAACCTAAGC
M	atcggccggc

Table 2 18 Synthesized Sequences	
Synthesized DNA fragment	DNA sequence (from 5' to 3')
Mv_1	atcggccggcACCTGTACTGATCG
$\overline{v_1M}$	gccggccgatCGATCAGTACAGGT
Mv_2	atcggccggcATGCTGAATGGCCT
$\overline{v_2M}$	gccggccgatAGGCCATTACAGCAT
...	...
Mv_6	atcggccggcGTCACGAACCTAAGC
$\overline{v_6M}$	gccggccgatGCTTAGTTCTGTGAC
$\begin{bmatrix} M \\ \overline{M} \end{bmatrix} \begin{pmatrix} v_1 \\ \lambda \end{pmatrix}$	atcggccggcACCTGTACTGATCG tagccggccg
$\begin{bmatrix} M \\ \overline{M} \end{bmatrix} \begin{pmatrix} v_2 \\ \lambda \end{pmatrix}$	atcggccggcATGCTGAATGGCCT tagccggccg
...	...
$\begin{bmatrix} M \\ \overline{M} \end{bmatrix} \begin{pmatrix} v_6 \\ \lambda \end{pmatrix}$	atcggccggcGTCACGAACCTAAGC tagccggccg

5 Conclusion and Remark

In this paper, we introduced a DNA algorithm for solving the graph coloring problem based on sticker system and deletion one. Compared with those approaches reported^[6-11], we need encode and synthesize less DNA sequences. In our algorithm, we need encode $n + 1$ oligonucleotides denoting $n + 1$ symbols v_1, v_2, \dots, v_n, M respectively and synthesize $3n$ DNA sequences $Mv_i, \overline{v_iM}$ and $\begin{bmatrix} M \\ \overline{M} \end{bmatrix} \begin{pmatrix} v_i \\ \lambda \end{pmatrix}$, $i = 1, 2, \dots, n$. However, in [6], Liu et al. must encode $2n + 1$ oligonucleotides representing n vertex-symbols N_1, N_2, \dots, N_n , n color-symbols C_1, C_2, \dots, C_n and one artificial vertex-symbol N_{n+1} re-

spectively, synthesize n^2 DNA sequences $N_iC_jN_{i+1}$ for odd i and $\overline{N_{i+1}C_jN_i}$ for even i , $j = 1, 2, \dots, n$; In [7], Liu et al. must encode $n + k$ oligonucleotides representing n vertex-symbols v_1, v_2, \dots, v_n and k color-symbols C_1, C_2, \dots, C_k respectively and synthesize nk DNA sequences, where k is the used color number. Furthermore, for any nonempty graph G , determining its chromatic number $\chi(G)$ need transfer the DNA algorithm of [7] $\chi(G) - 1$ times, namely, $k = 2, 3, \dots, \chi(G)$. In [8], Liu et al. must encode $2n + 1$ oligonucleotides denoting $n + 1$ order-symbols p_1, p_2, \dots, p_{n+1} and n vertex-symbols v_1, v_2, \dots, v_n respectively and synthesize n^2 DNA sequences $p_iv_j, i, j = 1, 2, \dots, n$. In [9], Xu Jin et al. must encode k oligonucleotides denoting k color-symbols and synthesize k^n DNA sequences representing all the possible colorings. In [10], Amos et al. must encode $2n$ oligonucleotides denoting n color-symbols and n position-symbols, synthesize nk DNA sequences representing all the possible colorings. In [11], Gao Lin and Xu Jin must encode $2n + 1$ oligonucleotides representing $n + 1$ order-symbols P_1, P_2, \dots, P_{n+1} and n value-symbols V_1, V_2, \dots, V_n respectively, synthesize n^2 DNA sequences $P_iV_jP_{i+1}$ for odd i and $\overline{P_{i+1}V_jP_i}$ for even $i, j = 1, 2, \dots, n$. We make comparisons with references [6]-[11] in Table 3.

Table 3 Comparisons with References [6]-[11]		
Algorithms	Encoded sequences	Synthesized sequences
Our alorithm	$n + 1$	$3n$
Reference [6]	$2n + 1$	n^2
Reference [7]	$n + k$	nk
Reference [8]	$2n + 1$	n^2
Reference [9]	k	k^n
Reference [10]	$2n$	nk
Reference [11]	$2n + 1$	n^2

References

[1] Kari L, Păun Gh, Rozenberg G, Salomaa A, Yu S. DNA computing, sticker systems, and universality. *Acta Informatica*, 1998, 35(5): 401-420

[2] Păun Gh, Rozenberg G. Sticker systems. *Theoretical Computer Science*, 1998, 204(1-2): 183-203

[3] Sakakibara Y, Kobayashi S. Sticker systems with complex structures. *Soft Computing*, 2001, 5(2): 114-120

[4] Kari L, Thierrin G. Contextual insertion/deletion and computability. *Information and Computation*, 1996, 131(1): 47-61

[5] Kari L. On insertion and deletion in formal language[Ph. D. dissertation]. University of Turku, Turku, 1991

[6] Liu Ya-Chun, Xu Jin, Pan Lin-Qiang, Wang Shi-Ying. DNA solution of a graph coloring problem. *Journal of Chemical Information and Computer Sciences*, 2002, 42(3): 524-528

- [7] Liu Wen-Bin, Zhang Feng-Yue, Xu Jin. DNA algorithm for the graph coloring problem. *Journal of Chemical Information and Computer Sciences*, 2002, 42(5): 1176-1178
- [8] Liu Xiao-Ming, Yin Jian-Wei, Jwo Jung-Sing et al. A DNA-based genetic algorithm implementation for graph coloring problem//*Proceedings of the International Conference on Intelligent Computing (ICIC05)*, Hefei, 2005:99-108
- [9] Xu Jin, Qiang Xiao-Li, Fang Gang et al. A DNA computer model for solving vertex coloring problem. *Chinese Science Bulletin*, 2006, 51(20): 2541-2549
- [10] Amos M et al. Error-resistant implementation of DNA computations//*Proceedings of the 2nd Annual Meeting on DNA Based Computing*, Princeton, 1996, 44: 151-161
- [11] Gao Lin, Xu Jin. A DNA algorithm for graph vertex coloring problem. *Acta Electronic Sinica*, 2003, 31(4): 494-496
- [12] Croitoru C, Lunchian H, Gheorghies O, Apetrei A. A new genetic graph coloring heuristic//*Proceedings of the Computational Symposium on Graph Coloring and its Generalizations*. Iasi, Romania, 2002: 63-74
- [13] Păun Gh, Rozenberg G, Salomaa A. *DNA Computing: New Computing Paradigms*. Berlin: Springer, 1998



WANG Shu-Dong, born in 1973, Ph. D., associate professor. Her research interests include biomolecular computing, graph theory and its applications etc. Her current research focuses on DNA computing models and applications.