MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **96** (2026) 75–98
ISSN: 0340–6253
doi: 10.46793/match.96-1.26825

The Calculation of Antiparallel Oriented Links Modeling DNA Polyhedrons

Xingmin Guo[a], Yifei Duan[b], Shuya Liu[a,*]

[a] *Key Laboratory of Colloid and Interface Chemistry, Shandong University, Jinan 250018, P. R. China*

[b] *Department of Statistics, Columbia University, New York, NY 10027, USA*

202312131@mail.sdu.edu.cn, yd2773@columbia.edu, liushuya@sdu.edu.cn

(Received October 13, 2025)

Abstract

How to determine the number and folding pathway of ssDNA strands assembling DNA polyhedrons are the fundamental problems in DNA nanoassembly. In this paper, the related mathematical problems are presented by introducing antiparallel oriented links (AO links) to describe the topological structures of DNA polyhedrons, that is, how to give all AO link diagrams based on any 2-connected plane graph and their link components. We demonstrate that each AO link diagram must be isotopy equivalent to a special AO link diagram with even tangle edges by defining "Node Move" operations on vertex nodes. By further giving the relationship between AO link diagrams and edge-weighted plane graphs, two algorithms are established to calculate all AO link diagrams and their link components. Based on these two algorithms, a software "AO link" has been developed in Fortran language, giving all AO link diagrams and their link components only by inputting a planar graph. Also, the other three versions of the software are established to eliminate isotopy classes of AO link diagrams. This work not only reveals some important properties of AO link diagrams but also provides an important tool for related chemical and mathematical problems.

*Corresponding author.

1 Introduction

In knot theory, a knot is a closed simple curve embedded in the 3D space and a link is a finite disjoint collection of knots [1]. With the rapid development of DNA nanotechnology, many mathematical knots and links have been realized in the nanostructures of DNA polyhedra [2, 3]. For example, some polyhedral knots were formed by folding a long DNA single strand, such as DNA tetrahedron [4], DNA triangular prism [5], DNA pyramid [6], DNA Pentagonal pyramid [7] and so on. Also, some covalently closed polyhedral catenanes were assembled from more than one single DNA strand, such as DNA tetrahedra [8, 9], DNA cube [10], DNA Triangular bipyramid [11], DNA Octahedra [12, 13] and so on. Obviously, the number and entanglement of ssDNA strands plays an important role in the formation of the structure of DNA polyhedra. Therefore, how to determine the number and folding pathway of ssDNA strands assembling DNA polyhedrons become the fundamental problems in DNA nanoassembly. To address it, some mathematical methods have been employed to characterize the topological structures of DNA polyhedron.

A DNA polyhedron can be represented as a compact orientable surface (called a thickened graph) by treating each double helix edge as a band having oriented boundary curves. Jonoska and Saito investigate the maximum and minimum numbers of ssDNA strands forming DNA polyhedrons with double helical edges by using the boundary curves of the thickened graph [14]. Also, Jonoska, Seeman and Wu prove that every connected multigraph permits a thickened graph with one boundary curve [15]. In fact, one boundary curve can be defined as an antiparallel strong trace of the graph such that no edge is traversed more than once in each direction. Fijavž, Pisanski and Rus characterize the graphs which admit parallel and antiparallel strong traces [16]. Then, Cheng, Deng and Diao further show that the strong traces of certain 2-connected plane graphs can be obtained by using thickened graphs constructed with only two types of vertex junctions [17]. These works provide some possible approaches to describing some DNA polyhedrons folded by one ssDNA strand.

Polyhedral links, the interlinked and interlocked architectures in poly-

hedral shape, have also served as the important mathematical models for describing the structural properties of DNA polyhedra [18–26]. In these works, some polyhedral knots based on platonic polyhedra and truncated polyhedra have been constructed by minimizing the number of link components [25, 26]. In fact, each polyhedral knot determines an antiparallel strong trace of the polyhedron. Meanwhile, all polyhedral links based on tetrahedra [27], trigonal prism [28,29], trigonal bipyramid [30,31] and octahedra [32] have been calculated by considering the antiparallel orientation of each tangle edge. These results provide all topological structures based on these four polyhedrons when they are assembled from one or more ssDNA strands. However, these works need all antiparallel orientations of polyhedral links determined in advance. This is a very tricky job to avoid the conflict orientation produced by any two adjacent tangle edges.

In this paper, antiparallel oriented links are defined to characterize the topological structures of DNA polyhedrons. First, we show that each AO link diagram can be continuously deformed into a special AO link diagram having even tangle edges by introducing an isotopy transform "Node Move" to a series of vertex nodes. Then, we establish the relationship between AO link diagrams and edge-weighted plane graphs by further exposing the properties of AO link diagrams. Based these results, two algorithms have been established for calculating all AO link diagrams based on a 2-connected plane graph and their link components. Also, a software "AO link" has been developed in Fortran language based on these two algorithms, which can give all AO link diagrams and their link components only by inputting a planar graph. Meanwhile, the other three versions of the software are established to eliminate some isomorphic weighted graphs leading to the isotopy link diagrams. As examples, all AO link diagrams based on some common polyhedrons including pyramids, prisms, bipyramids, dodecahedra and icosahedra and their link components have been calculated by our software. Our work not only reveals some important properties of AO link diagrams but also provides a tool to give all topological structures for DNA polyhedrons as well as the antiparallel strong trace and the upper embeddability for a plane graph.

2 Defining AO link diagrams

Some basic concepts, terminology and notation are given in advance.

A 2-tangle is two strands twisting against each other in antiparallel directions. A 2-tangle T have four endpoints in the plane, named as NW, NE, SW and SE, as indicated in Fig. 1(a). The NW and SW ends are defined as head of T, and the NE and SE ends are defined as tail of T. The 2-tangle T is called positive or negative according to the SW end entering or exiting T. *The reverse of* T, denoted by $-T$, is obtained from T by reversing the direction of each strand. In Fig. 1(b), the 2-tangles T_0, T_1 and T_2 are all positive, and their reverses $-T_0$, $-T_1$ and $-T_2$ are all negative.

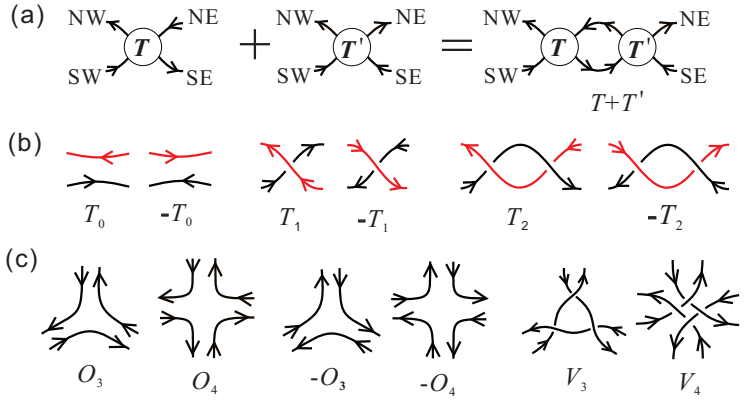

Figure 1. (a) The sum for the 2-tangles T and T' surrounded respectively by a circle; (b) The 2-tangles T_0, T_1 and T_2, and their reverses $-T_0$, $-T_1$ and $-T_2$; (c) The 0-node 0_k, its reverse -0_k, and the k-node V_k for $k = 3, 4$.

A k-tangle is composed of k ($k \in \mathbb{Z}^+ > 2$) strands such that each strand is oriented counterclockwise. A k-tangle is called *a k-node*, denoted by V_k, if each strand crosses the other two strands alternately. A k-tangle having no crossing, denoted by 0_k, is called *0-node*, as illustrated in Fig. 1(c). *The reverse of a k-tangle* 0_k, denoted by -0_k, is obtained by reversing the direction of each strand. *The sum of two 2-tangles* T and T', denoted by $T+T'$, is the 2-tangle obtained by gluing the NE and SE end of T together

with the NW and SW end of T' according to the orientation of their ends (Fig. 1(a)). A 2-tangle having $2n$ ($n \in \mathbb{N}$) crossings, denoted by T_{2n}, is called *even*. Similarly, *an odd 2-tangle* T_{2n+1} is a 2-tangle having $2n+1$ crossings. Note that T_{2n+1} is coincident with $-T_{2n+1}$ by a 180-degree rotation, and they are distinguished by their head or tail.

Let L be any given oriented alternating link. Without loss of generality, we can suppose that L be reduced and have no nugatory crossing. Given any oriented alternating link diagram D of L, there must be a 2-connected plane graph G' corresponding to D according to the checkerboard coloring of D (Fig. 2) [33]. For the graph G', each maximum path with all interior vertices having degree 2 is replaced with an edge, and the resulting graph G is a 2-connected plane graph having no vertex of degree 2. If each edge e of G is exactly corresponding to a 2-tangle T_e, the link diagram D is called *an antiparallel oriented link diagram* or *an AO link diagram* (corresponding to the plane graph G), and the link L is called *an AO link*. In particular, the AO link L is called *an AO polyhedral link* if G is a 3-connected plane graph.

Conversely, given any 2-connected plane graph G, each edge is replaced by a 2-tangle, and connect two adjacent endpoints of two tangles along each face of G, the resulting oriented link diagram is exactly an AO link diagram $D(G)$. If G has a vertex v of degree 2, the maximum path P containing v is corresponding to a 2-tangle in $D(G)$, which are the sum of all 2-tangles corresponding to all edge of P. Evidently, $D(G)$ also can be constructed from the plane graph resulted by replacing the path with an edge. Hence, we can suppose that *the plane graphs involved in this paper are all 2-connected and have no vertex of degree 2.* According to our construction method, any AO link diagram can be obtained from a 2-connected plane graph G, as illustrated in Fig. 2. In addition, the k-tangle of $D(G)$ corresponding to each vertex of G is called *the vertex node* of $D(G)$. In Fig. 3, $D(G)$ have two 0_3 and two -0_3 as vertex nodes.

If G is a plane graph of a polyhedron P, the diagram $D(G)$ can be transformed into an AO polyhedral link $L(G)$ by recovering P from G according to the plane projection from the inside out. In fact, a polyhedron has identical faces in any plane drawing since any 3-connected planar graph

can be uniquely embeddable on the sphere. Therefore, AO polyhedral link diagrams don't depend on any plane graph of the polyhedron P, which can be constructed from any plane graph of P.

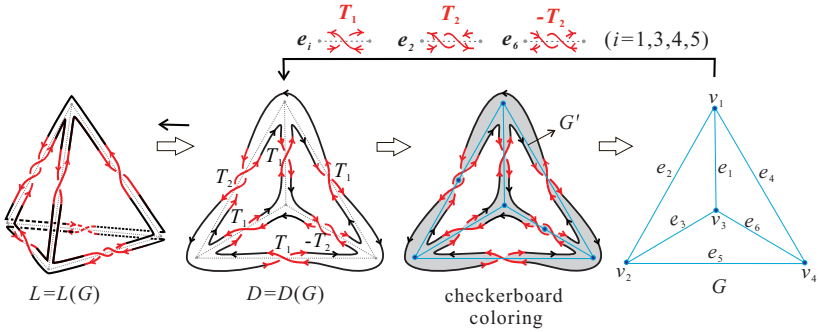

Figure 2. The link diagram D of the link L determines a plane graph G according to the checkerboard coloring of D; Conversely, an AO link diagram $D(G)$ can be obtained from G by replacing each edge e_i with the 2-tangle T_1, T_2 or $-T_2 (1 \leq i \leq 6)$, and the AO link $L(G)$ can be recovered from $D(G)$.

3 Some properties of AO link diagrams

In this section, some properties of AO link diagrams will be given. Firstly, we can obtain the following lemma according to the construction of AO link diagrams.

Lemma 1. *Let $D(G)$ be any AO link diagram corresponding to the plane graph G. Any vertex node n_v of $D(G)$ must be a 0-node 0_k or its reverse -0_k, where k is the degree of the corresponding vertex v of G.*

Proof: According to the construction of $D(G)$, the vertex node n_v is composed of some disjointed arcs such that each arc connects two adjacent endpoints of two tangle edges. The number of these arcs are exactly equal to the number of these edges incident to the vertex v in G. Therefore, the vertex node n_v is composed of k arcs such that any two arcs have no any crossing according to the construction of AO link diagrams.

Also, for the vertex node n_v, each arc connecting two adjacent endpoints of two tangle edges will inherit a clockwise or anticlockwise orientation. If an arc l of the vertex node n_v is oriented clockwise, two arcs adjacent to l must be oriented clockwise due to the antiparallel orientations of two tangle edges connected to l. And so on, each arc of n_v is oriented clockwise. Similarly, we can show that each arc of n_v is oriented anticlockwise if an arc of n_v is oriented anticlockwise. Hence the vertex node n_v must be 0-node 0_k or its reverse -0_k. □

An AO link diagram $D(G)$ only having positive and even 2-tangle edges, denoted by $D^E(G)$, is called *a positive EAO link diagram*, as shown in Fig.3(b). In fact, a positive EAO link diagram $D^E(G)$ must exist for any plane graph G such that each link component of $D^E(G)$ walks clockwise along exactly a face of G. Hence each vertex node of $D^E(G)$ must be a 0-node 0_k according to lemma 1.

In any given AO link diagram, a node -0_k is transformed into a k-node V_k when a crossing of each 2-tangle edge incident to the -0_k is moved continuously into it, as illustrated in Fig. 3(a). This transformation, called *Node Move*, change the node -0_k to a k-node V_k. Evidently, Node Move is an isotopy transform, and the link type of $D(G)$ is unchanged under Node Moves. An oriented link diagram $D^s(G)$ is called a *special AO link diagram* or *SAO link diagram* if there exists a plane graph G such that each edge and each vertex are corresponding to a 2-tangle edge and a vertex node 0_k or V_k of $D^s(G)$, respectively. Also, each such k-node V_k is called a vertex node of $D^s(G)$. A SAO link diagram $D^s(G)$ based on a tetrahedron G is given in Fig.3(b), which is obtained from an AO link diagram $D(G)$ by applying Node Move to each vertex node -0_3. Then we have the theorem below.

Theorem 1. *Each AO link diagram is isotopy equivalent to a SAO link diagram, and each SAO link diagram only have positive even 2-tangle edges.*

Proof: Given any AO link diagram $D(G)$, $D(G)$ can be obtained from a plane graph G by replacing each edge with a 2-tangle T. Hence each vertex node of $D(G)$ is a 0-node 0_k or its reverse -0_k according to lemma 1. For any vertex node -0_k of $D(G)$, -0_k must be connected to the head

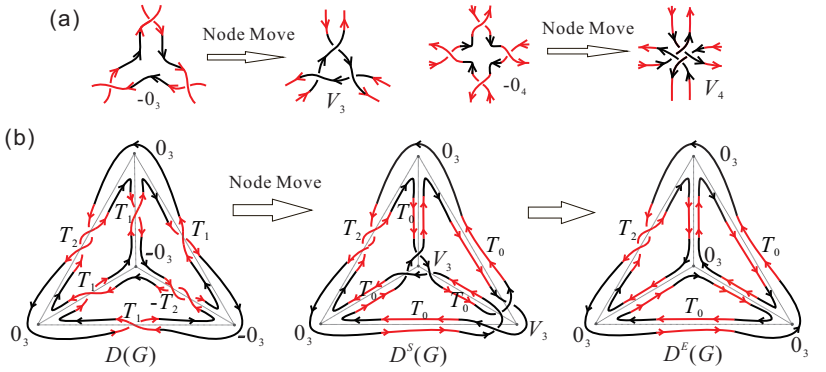

Figure 3. (1) The Node Move is respectively applied to the vertex nodes -0_3 and -0_4, and each pair of red arcs indicates a part of a 2-tangle edge. (2) The AO link diagram $D(G)$, the SAO link diagram $D^S(G)$ obtained from $D(G)$ by applying the Node Move to each -0_3, and the positive EAO link diagram $D^E(G)$ obtained from $D^S(G)$ by replacing each V_3 with a 3-tangle 0_3.

of a negative 2-tangle $-T_{2m+1}$, the tail of a positive 2-tangle T_{2m+1}, or the head or tail of a negative even 2-tangle $-T_{2m}$ according to its orientation. Also, we have

$T_{2m+1} = T_{2m} + T_1$, $-T_{2m+1} = -T_1 + T_{2m}$ and $-T_{2m} = -T_1 + T_{2m} + T_1$, where the" T_1 " or "$-T_1$ " end in T_{2m+1}, $-T_{2m+1}$ and $-T_{2m}$ is connected to the vertex node -0_k. Hence applying Node Move to the vertex node -0_k, the vertex node -0_k is changed into a k-node V_k. Meanwhile, each odd tangle is changed into an even tangle, and each even tangle $-T_{2m}$ is changed into the odd tangle $-T_{2m+1}$. Repeatedly apply Node Move to each vertex node -0_k of $D(G)$ until each vertex node is 0_k or V_k. Evidently, the resulting link diagram $D^s(G)$ is a SAO link diagram, which is isotopy equivalent to $D(G)$.

Note that each edge of $D^s(G)$ must be a positive even 2-tangle. If an odd or negative tangle appears in $D^s(G)$, the tangle may be the positive tangle T_{2m+1} or the negative tangle $-T_{2m+1}$ or $-T_{2m}$. According to the orientation of the 2-tangle, there is at least one node -0_k that is connected to the head of $-T_{2m+1}$, the tail ofT_{2m+1}, or the head or tail of $-T_{2m}$. This contradicts that $D^s(G)$ have no node -0_k. The prove is finished. □

The reverse of a link diagram D, denoted by $-D$, is also an AO link diagram that are obtained by reversing the orientation of each component. Hence the reverse of a SAO link diagram only contains vertex nodes -0_k or $-V_k$ and negative even 2-tangle edges. Hence we can obtain the following corollary by the theorem 1.

Corollary. *(a) Each AO link diagram is isotopy equivalent to the reverse of a SAO link diagram; (b) Each DNA polyhedron can be designed as a SAO link having DNA double helical edges of the same length.*

Also, for any given AO link diagram $D(G)$, we have the lemma below.

Lemma 2. *Each oriented link diagram obtained from $D(G)$ by reversing the orientations of some link components is not an AO link diagram except for $-D(G)$.*

Proof: Let $D^O(G)$ be an AO link diagram obtained from $D(G)$ by reversing the orientations of some link components. By the theorem 1, $D(G)$ is isotopy equivalent to a SAO link diagram $D^{s1}(G)$. Then $D^O(G)$ is isotopy equivalent to the oriented link diagram $D^{s2}(G)$ obtained from $D^{s1}(G)$ by reversing the orientations of the corresponding link components. If $D^{s2}(G)$ have a non 2-tangle edge, $D^{s2}(G)$ must not be an AO link diagram. Otherwise, $D^{s2}(G)$ must have a negative 2-tangle edge $-T_{2n}$. For $D^{s2}(G)$, each vertex node connected to $-T_{2n}$ must be -0_k or $-V_k$, and each 2-tangle edge connected to the -0_k or $-V_k$ must be negative due to their even crossing number. And so on, any vertex node can reach to the vertex node -0_k or $-V_k$ in $D^{s2}(G)$ by a serial of negative 2-tangle edges and vertex nodes -0_k or $-V_k$ when G is connected. Therefore, each vertex node in $D^{s2}(G)$ must be the node -0_k or $-V_k$, and each 2-tangle edge is negative. Then $D^{s2}(G)$ is the reverse of $D^{s1}(G)$ and $D^O(G)$ is isotopy equivalent to the reverse of $D^{s2}(G)$. □

According to theorem 1, any AO link diagram $D(G)$ can be transformed into a SAO link diagram $D^s(G)$ by changing each vertex node -0_k to a k-node V_k. Collecting each vertex of G corresponding to each node -0_k gives a subset S of $V(G)$. Conversely, given any subset S, a SAO link diagram $D^s(G)$ can be obtained from a positive EAO link diagram $D^E(G)$

by replacing each corresponding 0-node 0_k with a k-node V_k. Also, an AO link diagram $D(G)$ can be obtained from $D^s(G)$ by applying the inverse transformation of Node Move to each vertex node V_k according to the theorem 1. Hence the set S can give the diagram $D(G)$, and we have the lemma 3.

Lemma 3. *All AO link diagrams obtained from any given plane graph G and all subsets of the vertex set $V(G)$ can determine each other if only considering the parity of each 2-tangle edge.*

Proof: Let $D_1(G)$ and $D_2(G)$ be two AO link diagrams given by any subset S of $V(G)$. We only need to show that $D_1(G)$ and $D_2(G)$ have the same parity for any two corresponding 2-tangle edges. By theorem 1, the AO link diagrams $D_1(G)$ and $D_2(G)$ are isotopy equivalent to two SAO link diagrams $D_{10}(G)$ and $D_{20}(G)$, respectively. Then $D_{10}(G)$ and $D_{20}(G)$ have the same vertex nodes and have the positive even 2-tangle edges. Through applying the inverse transformation of Node Move to one same vertex node V_k in $D_{10}(G)$ and $D_{20}(G)$, all 2-tangle edges connected to V_k are changed into odd 2-tangles, and the parity of the remain 2-tangle edges is unchanged in $D_{10}(G)$ and $D_{20}(G)$. Repeatedly applying this operation to the remaining k-nodes, $D_1(G)$ and $D_2(G)$ can recovers respectively from $D_{10}(G)$ and $D_{20}(G)$. Accordingly, any two corresponding 2-tangles in $D_1(G)$ and $D_2(G)$ have the same parity. □

According to the lemma 3, all AO link diagrams only having one or two crossings for each 2-tangle edge can be given by considering all subsets of the vertex set $V(G)$. These AO link diagrams can generate all AO link diagrams by changing the crossing number according to the parity of each 2-tangle. Hereafter, for convenience, we suppose that all AO link diagrams only having one or two crossings on each 2-tangle edge.

Let P be a 2-connected simple planar graph without 2-degree vertices that is a triple consisting of a vertex set $V(P)$, an edge set $E(P)$, and an incident relation $R(P)$ that associates with each edge two vertices called its endpoints. Let G be a plane graph of P, which have the same vertex set, the edge set and the incident relation as P.

For any subset S of $V(G)$, an edge e of the plane graph G will be

weighted with 2 if the edge e is incident with exactly one vertex of S. Otherwise, the edge e will be weighted with 1. The resulting weighted graph, denoted by $W_s(G)$, depend only on the subset S and the incident relation $R(G)$, not on the plane graph G. Hence the weighted graph can be naturally extended to the planar graph P from G.

For the link diagram $D(G)$ determined by the set S, each vertex in S is corresponding to a vertex node -0_k of $D(G)$. In $D(G)$, each 2-tangle edge connected exactly to one vertex node -0_k must have the odd-crossing number and the remaining 2-tangles all have the even-crossing number. Hence, in $W_s(G)$, each edge corresponding to an odd 2-tangle edge must have weight 1 and each edge corresponding to an even 2-tangle edge must have weight 2. Hence, we have the theorem 2.

Theorem 2. *Any AO link diagram $D(G)$ and its reverse $-D(G)$ determine an identical weighted graph $W_s(G)$. Conversely, any weighted graph $W_s(G)$ also determine two AO link diagrams $D(G)$ and $-D(G)$ for a subset S of the vertex set $V(G)$.*

Proof: There are two subsets S and S^c of $V(G)$ determined respectively by $D(G)$ and $-D(G)$ according to the lemma 3. Let $W_s(G)$ and $W_s^c(G)$ be the weighted graphs given respectively by the subsets S and S^c, respectively. Note that the subsets S and S^c are corresponding to all -0_k nodes of $D(G)$ and $-D(G)$, respectively. Then S and S^c must be complementary, that is $S \cup S^c = V(G)$. For each edge e of G, if S exactly contain one endpoint of e, S^c exactly contain the other endpoint of e. Hence the edge e has the same weight 2 in $W_s(G)$ and $W_s^c(G)$. Otherwise, either S or S^c contain both endpoints of e. Also, the edge e has the same weight 1 in $W_s(G)$ and $W_s^c(G)$. Hence $W_s(G)$ and $W_s^c(G)$ are the same weighted graph. Also, from the point of view of link diagrams, each edge must have the same weight in $W_s(G)$ and $W_s^c(G)$ since each 2-tangle have the same parity in $D(G)$ and $-D(G)$. Conversely, given the weighted graph $W_s(G)$, by ignoring the orientation, an edge of G is replaced by an odd 2-tangle if its weight is 2 in $W_s(G)$. Otherwise, this edge is replaced by an even 2-tangle. According to the construction method in the section 2, an unoriented link diagram can be obtained. By lemma 3, the link diagram

must allow two AO link diagrams $D(G)$ and $-D(G)$ determined by the subset S and S^c. In fact, $D(G)$ can be easily obtained from the unoriented link diagram according to the subset S if each corresponding vertex node is assigned to -0_k and the remaining vertex nodes are assigned to 0_k. Also, by lemma 2, the link diagram only allows an AO link diagram $D(G)$ and its reverse $-D(G)$. Hence the AO link diagrams $D(G)$ and $-D(G)$ are determined completely by the weighted graph $W_s(G)$. □

4 Two Algorithms for AO link diagrams

In this section, the number of elements of a set $*$ is denoted by $|*|$. Let P be a simple 2-connected planar graph defined in the section 3. Let $n = |V(P)|$ and $m = |E(P)|$ be the number of vertices and edges of P, respectively. Let n_f be the number of faces of P such that $n_f = m - n + 2$. For any subset S of $V(P)$, an incidence matrix of the weighted graph $W_s(P)$, called *a weighted incidence matrix*, can be obtained from the incidence matrix of P by replacing 1 with the weight $w(e)$ for each edge e.

4.1 An algorithm for generating AO link diagrams

According to theorem 2, all AO link diagrams and their reverses based on the plane graph G of P can be calculated from the incidence matrix M of P by using the algorithm below.

Algorithm 1.

Input: *A $n \times m$ incidence matrix M of P.*

Idea: *Generating the n-dimensional arrays of 2^n such that each component is 0 or 1. These arrays are numbered as $X_1, X_2, \cdots, X_{2^n}$.*

For each array X_i ($1 \leq i \leq 2^n$), each nonzero value w in the j-th row of M is converted to 1 from 2 or to 2 from 1 if the j-th component of X_i is 1. Otherwise, the value w is unchanged in the j-th row. When j takes all components of X_i, a weighted incidence matrix M_i of P is obtained. Collecting all weighted incidence matrix M_i forms the set $\mathcal{M}$.

Then delete any isomorphic weighted graph. Firstly, all weighted incidence matrix M_i in $\mathcal{M}$ are divided into finitely many sets G_k, ($k \leq m$)

according to the number k of the columns containing 2 in M_i. Then delete any equivalent matrices in G_k, and merge all sets G_k to replace the set $\mathcal{M}$.

Initialization: *i=1, j=1, k=1, $\mathcal{M} = \phi$*

Iteration:

1) For $i \leq 2^n$, we have the following iteration. For $j \leq n$, each nonzero value w in the j-th row of M is converted to 1 from 2 or to 2 from 1 if the j-th component of X_i is 1. Otherwise, the value w is unchanged in the j-th row. Then the index j turns to $j+1$ for the next loop, and iterating until $j = n+1$. The resulting weighted incidence matrix M_i is collected to the set $\mathcal{M}$. Then the index i turns to $i+1$ for the next loop, and iterating until $i = 2n+1$. All weighted incidence matrix M_i are collected to $\mathcal{M}$, and they are numbered from 1 to 2^n.

2) Through calculating the number of the columns containing 2 in each weighted incidence matrix in $\mathcal{M}$, all matrixes of $\mathcal{M}$ are divided into the sets G_k $(0 \leq k \leq s)$ such that any two matrixes have the same number of the columns containing 2 in G_k.

For $k \leq s$, we have the following iterate on the set G_k.

For $i < |G_k|$, through the permutation of the row vectors of the i-th matrix in G_k, the resulting matrices of n! are compared with each matrix with marked numbers higher than i in G_k. If two matrices have exactly the same column vectors, then delete the matrix with the marked number higher than i from G_k. Subsequently, renumber those surviving matrices in order. Then the index i turns to $i+1$ for the next loop, and iterating until $i = |G_k|$. If $i = |G_k|$, this iteration stops and the index k advances. The iteration on G_k continues until $k = s+1$. All sets G_k $(0 \leq k \leq s)$ are merged to replace all elements of the set $\mathcal{M}$. Output the set $\mathcal{M}$.

**2) Suppose that G is a polyhedron with the symmetry group $\mathcal{K}$ generated by some symmetry operations. Each weighted incidence matrix M_i can induce an ordered array Y_i of edges such that the weight of the k-th edge is taken as the k-th component for $1 \leq k \leq m$. These ordered arrays are collected to give the set $\mathcal{Y}$, which inherit a marked number from each weighted incidence matrix in $\mathcal{M}$.*

For $i < |\mathcal{Y}|$, through applying each symmetry operation of $\mathcal{K}$ to the

ordered array Y_i, the resulting ordered arrays of $|\mathcal{K}|$ are compared with each ordered array Y_j with marked number j higher than i in $\mathcal{Y}$. If the ordered array Y_j is the same as one of these arrays of $|\mathcal{K}|$, Y_j is removed from $\mathcal{Y}$ and the corresponding matrix M_j is deleted from the set $\mathcal{M}$. At last, renumber those surviving arrays and matrices in their original order. Then the index i turns to $i+1$ for the next loop, and iterating until $i = |\mathcal{Y}|$. Output the set $\mathcal{M}$.

According to the Algorithm 1, the weighted incidence matrixes of all weighted graphs for the planar graph P can be generated in the part 1) of iteration process. Also, the part 2) of iteration process are devoted to deleting the isomorphic weighted graphs through the equivalence of their weighted incidence matrixes. If the graph P is a polyhedron with high symmetry group $\mathcal{K}$, we will use the algorithm of the part *2) to eliminate the equivalent weighted incidence matrixes by using all symmetric operations from $\mathcal{K}$. Without considering the isomorphism of weighted graphs, Algorithm 1 primarily involves generating both n-dimensional arrays and weighted incidence matrixes, resulting in a time complexity of $O(n \cdot 2^n)$. Consequently, we estimate that Algorithm 1 can be applied to a planar graph with approximately n=26 vertices when the maximum running time is set to one second.

Note that all weighted incidence matrixes can be computed by the incidence matrix M of P according to the Algorithm 1. Hence any two plane graphs of P have the same weighted incidence matrixes. Once a plane graph G of P is given, then according to theorem 2, all weighted graphs of G determined by all weighted incidence matrices can give all AO link diagram based on G. When P is a polyhedron, the construction of AO link diagrams only depends on P since all faces don't change in any plane graph of P.

4.2 An algorithm for calculating link components

For any 2-connected plane graph G, each cycle bounded by each face is called a face cycle of G. Here each face of G is given by recording each edge clockwise or anticlockwise along its face cycle. Evidently, a plane graph

can be determined by giving all faces. For any 2-connected planar graph P, the incidence matrix M can be used to generate all faces by simplifying the planarity algorithm [34], that exactly give a plane graph of P.

For the plane graph G, each edge e is shared by two face cycles, hence there are two adjacent edges of e on each face circle. These four adjacent edges, denoted by e_s, e_a, e_o and e_d, are called "the starting edge", "the adjacent edge", "the opposite edge" and "the diagonal edge" according to the position relative to e (Fig. 4). The four adjacent edges together with the edge e forms four pairs of edges $[e, e_s]$, $[e, e_a]$, $[e, e_o]$ and $[e, e_d]$, respectively. In fact, each pair of edges describes one arc connecting two adjacent endpoints of two 2-tangle edges for an AO link diagram. Hence each pair of edges is also called an arc. Evidently, each link component can be recorded by some pairs of edges when walking along this component.

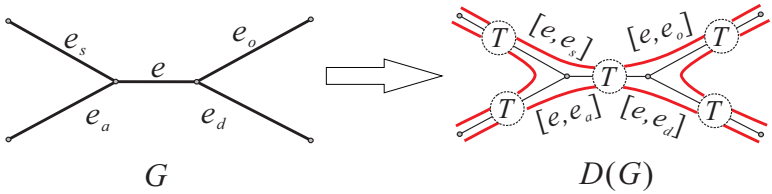

Figure 4. The edge e and its four adjacent edge e_s, e_a, e_o and e_d in the plane graph G (left graph); Four pairs of edges $[e, e_s]$, $[e, e_a]$, $[e, e_o]$ and $[e, e_d]$ respectively indicates four red arcs of an AO link diagram $D(G)$, where each arc connect two 2-tangles T surrounded respectively by a circle(right graph).

Note that the link components of any given AO link diagram $D(G)$ depend on the parity of the crossing number of each 2-tangle, not its orientation. Hence each link component of $D(G)$ can be calculated directly according to the corresponding weighted plane graph $W_s(G)$. For each edge e, the related four pairs of edges are divided into two sets. If $w(e) = 1$, two pairs of edges indicating two arcs, that are $[e, e_s]$ and $[e, e_o]$, form one 2-element set, and the other 2-element set are formed by the pairs of edges $[e, e_a]$ and $[e, e_d]$. Otherwise, the pairs of edges $[e, e_s]$ and $[e, e_d]$ form one 2-element set, and the pairs of edges $[e, e_o]$ and $[e, e_a]$ form the other 2-element set. Therefore, each such 2-element set can be given according

to the weight of each edge of $W_s(G)$, and any two 2-element sets will be further merged into a set of edge pairs if exactly two edge-pairs from these two 2-element sets contain an identical edge. At last, all pairs of edges are partitioned into k ($k \in \mathbb{Z}$) classes, and each link component of any AO link diagram is given as a class of edge pairs. Hence the link component of all AO link diagrams based on some plane graph of P can be calculated by using each weighted incidence matrix of P as follows.

Algorithm 2.

Input: *The incidence matrix $M(P)$ of a planar graph P, and the set $\mathcal{M}$. Collecting each weighted incidence matrix M_k of P ($1 \leq k \leq |\mathcal{M}|$).*

Idea: *The edge set $E(P)$ and the vertex set $V(P)$ obtained from $M(P)$ are collected for giving the face set $\mathcal{F}$.*

Firstly, find a face f of P. Starting from any vertex v of P, search all edges incident to v, and according to the other endpoint of each incident edge, search all edges incident to these endpoints. And so on, the search will stop until the vertex v appears again. This cycle starting and ending with v is a minimum cycle containing v, that must be a face f of P. Here the face f is represented as an ordered set of edges by recording each edge along the boundary. Add the face f and its unbound face to the set $\mathcal{F}$.

Secondly, find all faces for determining a plane graph of P. For the subgraph $P_{\mathcal{F}}$ induced by $E(P) - E(\mathcal{F})$ (called the fragments of P about $\mathcal{F}$), take any connected component g (i.e. a fragment) from $P_{\mathcal{F}}$, and find a path p from the component g such that only both endpoints of p belong to $V(\mathcal{F})$. Then the path p is added to a face f_a of $\mathcal{F}$ such that both endpoints of p belong to $V(f_a)$. The face f_a is split into two faces f_{a1} and f_{a2} by p. Delete f_a from $\mathcal{F}$ and add the faces f_{a1} and f_{a2} to $\mathcal{F}$. And so on, each face as an ordered set of edges is collected into $\mathcal{F}$, determining a plane graph G of P.

At last, give all link components of each link diagram based on G according to each weighted incidence matrix M_k in $\mathcal{M}$. For each edge e_i of $E(G)$, find two faces f_{i1} and f_{i2} containing e_i from $\mathcal{F}$. Two edges e_{is} and e_{io} adjacent to e_i on the face f_{i1}, and two edges e_{ia} and e_{id} adjacent to e_i on the face f_{i2} give an ordered four-bit array (e_{is}, e_{ia}, e_{id}, e_{io}) such that the edges e_{is}, e_{ia} share a common vertex. The edge e_i and its four

adjacent edges induces four pairs of edges, which are divided into two sets O_{i1} and O_{i2} according to the weight $w(e_i)$ in M_k and the obtained four-bit array. These two sets O_{i1} and O_{i2} are compared with each element O (i.e. a set collecting some pairs of edges) in the set $\mathcal{O}$. If the set O_{i1} or O_{i2} and O have a common edge, the set will be merged into the set O as a new set collecting some pairs of edges. Otherwise, this set is collected as a new element into $\mathcal{O}$. When each edge in $E(G)$ is taken, each link component of the link diagram determined by M_k will be given as an element in $\mathcal{O}$.

Initialization: *$\mathcal{F} = \phi$, $\mathcal{O} = \phi$ and $\mathcal{S} = \phi$.*

Iteration: *For $|\mathcal{F}| < n_f = m - n + 2$, the subgraph $P_{\mathcal{F}}$ induced by $E(P) - E(\mathcal{F})$ is given. Take any connected component g of $P_{\mathcal{F}}$, and find a path p from g such that only both endpoints of p belong to $V(\mathcal{F})$. The path p is added to a face f_a of $\mathcal{F}$ such that both endpoints of p belong to $V(f_a)$. The face f_a is split into two faces f_{a1} and f_{a2} by p. Delete f_a from $\mathcal{F}$ and add the faces f_{a1} and f_{a2} to $\mathcal{F}$. This iteration continues until $|\mathcal{F}| = n_f$. If $|\mathcal{F}| = n_f$, each face as an ordered set of edges has been collected into $\mathcal{F}$, which determine a plane graph G of P.*

For $k \leq |\mathcal{M}|$, we have the following iteration on the weighted incidence matrix M_k.

For $i \leq m$, we have the following iteration on $E(P)$. For each edge e_i of $E(P)$, find two faces f_{i1} and f_{i2} containing e_i from $\mathcal{F}$. Two edges e_{is} and e_{io} adjacent to e_i on the face f_{i1}, and two edges e_{ia} and e_{id} adjacent to e_i on the face f_{i2} give an ordered four-bit array (e_{is},e_{ia}, e_{io},e_{id}) such that the edges e_{is},e_{ia} share a common vertex. If $w(e_i) = 1$ in M_k, $[e_i, e_{is}]$ and $[e_i, e_{io}]$ are both collected into the set O_{i1}, and $[e_i, e_{ia}]$ and $[e_i, e_{id}]$ are collected into the set O_{i2}.Otherwise, $[e_i, e_{id}]$ and $[e_i, e_{is}]$ are collected into the set O_{i1}, and $[e_i, e_{ia}]$ and $[e_i, e_{io}]$ are collected into the set O_{i2}. Then the sets O_{i1} and O_{i2} are compared with each element $\mathcal{O}$ in the set O. If the set O_{i1} or O_{i2} and O have a common edge, it will be merged into the set O. Otherwise, the set will be collected into the set $\mathcal{O}$. The index i turns to $i + 1$ for the next loop, and iterating until $i = m + 1$. Each component of the link diagram determined by the matrix M_k can be given as an element in $\mathcal{O}$. The set $\mathcal{O}$ is copied as a new element into the set $\mathcal{S}$. Deleted all element of $\mathcal{O}$.

The index k advances and the above iteration continues until $k = |\mathcal{M}|+1$. All link components of each link diagram determined by all weighted incidence matrixes are collected into $\mathcal{S}$.

In Algorithm 2, all faces determining a plane graph G are randomly generated by the incidence matrix of P. Hence the plane graph G is also randomly given unless P is a polyhedron. If we need to calculate all link components of each AO link diagram based on a given plane graph H, each face of H can be entered directly instead of randomly generating all faces of some plane graph in Algorithm 2. Once a plane graph G is given, the complexity of Algorithm 2 depends primarily on the number of edges m and the number of faces n_f, and the length of the longest face cycle $m_{f_{max}}$ in G, yielding a worst-case complexity of $O(m^2 \cdot m_{f_{max}})$.

5 Software for calculating AO link diagrams

According to Algorithms 1 and 2, a computer software "AO link" has been developed in Fortran language, which can give all AO link diagrams based on any plane graph and their link components only by inputting an incidence matrix of the graph. The executable and input file of the software "AO link" are both given in *Supplementary material*. Note that the AO link diagrams obtained by the software "AO link" may contain many isotopic link diagrams resulted by the isomorphism of weighted graphs. The software "AO link" can be modified to another version "AO1 link" by adding the part 2) of Algorithm 1, which can eliminate all isomorphic weighted graphs based on any plane graph with less than 12 vertices. Take tetrahedra for example, the edges and vertices of tetrahedra G are notated in Fig.2. By inputting an incident matrix of G according to the subscripts of edges and vertices, there are three different weighted incident matrixes can be obtained by using the software "AO1 link" as follows.

"result_label= 1

matrix =

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

0 0 0 1 1 1

result_label= 2

matrix =

1 1 0 2 0 0

0 1 1 0 2 0

1 0 1 0 0 2

0 0 0 2 2 2

result_label = 3

matrix =

2 1 0 2 0 0

0 1 2 0 2 0

2 0 2 0 0 1

0 0 0 2 2 1 ”

The above three weighted incident matrixes determine three AO link diagrams, and the link diagram $D(G)$ in Fig.2 is determined by the matrix labeled as 3. According to these three matrixes, the software “AO1 link” can give each link component of three AO links as follows.

“ result_label = 1

one_branch = 1 2 2 3 3 1 1 2 //one_branch=one link component

one_branch = 1 4 4 6 6 1 1 4

one_branch = 2 4 4 5 5 2 2 4

one_branch = 3 5 5 6 6 3 3 5

branches_num = 4 //branches_num=the number of link components

result_label = 2

one_branch = 1 2 2 3 3 1 1 2

one_branch = 1 4 4 5 5 3 3 6 6 4 4 2 2 5 5 6 6 1 1

branches_num = 2

result_label = 3

one_branch = 1 2 2 3 3 6 6 5 5 2 2 4 4 6 6 1 1 2

one_branch = 1 4 4 5 5 3 3 1 1 4

branches_num = 2

branches_num = 2, results_num = 2

branches_num = 4, results_num = 1 ”

Each link component is given as a serial of the subscripts of edges of

G in above results. The first AO link diagram have 4 link components and the other two AO link diagrams have 2 link components. For all pyramids, prisms, and bipyramids with less than 12 vertices, all AO link diagrams based on these polyhedrons and their link components are given in *Supplementary material* by using the software "AO1 link". According to these results, the numbers of these AO link diagrams are given according to the number of their link components in table 1. Also, the total number of these diagrams are compared with the total number of all AO link diagrams obtained by using the software "AO link", as shown in table 1. Note that a tetrahedral link of four components [9], a 4-pyramid knot [6], a 3-prism knot [5], a 3-bipyramid link of six components [11], an octahedral link of eight components [12], and so on, have been successfully synthesized as DNA polyhedrons. In addition, the software "AO link" is also applicable to calculating the topological structures of the more complex DNA polyhedra [35], which can be obtained from polyhedra (or truncated polyhedra) with double or multiple edges according to our algorithm.

If the planar graph P is a polyhedron with high symmetry group K, the software "AO1 link" can be revised by replacing the part 2) with the part $*$ 2) according to the Algorithm 1, in order to eliminate the isomorphic weighted graphs by using all symmetric operations in K. For example, two software "Dodecahedra" and "Icosahedra" based on dodecahedra and icosahedra are established by modifying the software "AO link", and their executable files and input files are included in *Supplementary materials*. Both of dodecahedra and icosahedra belong to the point group I, which have 60 symmetric operations generated by 1 C_5 axis, 1 C_2 axis and 3 C_3 axes perpendicular to each other. Through respectively applying these symmetric operations to the edge sets of these two polyhedrons, two symmetry groups K_D and K_I base on their edge sets can be generated to eliminate the isomorphic weighted graphs. As a result, the total number of all AO link diagrams based on icosahedra are reduced to 9040 from 1048576 by using the symmetry groups K_D. Also, the total number for all AO link diagrams based on dodecahedra are reduced to 56 from 4096 by using the symmetry groups K_I.

In addition, the software "AO link" can also be used to check whether

Table 1. The list of the number of AO polyhedral links of k link components. N denotes the number of all AO link diagrams based on a polyhedron, and M denotes the number of all AO link diagrams obtained by deleting any isomorphic weighted graph.

Polyhedron \ k	1	2	3	4	5	6	7	8	10	12	20	M/N
tetrahedra		2		1								3 /16
4-pyramid	2		3	1								6 /32
5-pyramid		5		2		1						8 /64
6-pyramid	3		7		2		1					13 /128
3-prism	2		5		1							8 /64
cube		8		5		1						14 /256
5-prism	10		28		5		1					44 /1024
3-bipyramid		3		2		1						6 /32
Octahedra		1		3		1		1				6 /64
5-bipyramid		3		4		3		1	1			12 /128
Dodecahedra		5036		3536		434		31	2	1		9040/1048576
Icosahedra		2		12		13		13	8	5	1	56/4096

a planar graph allows an AO link of k components, further giving the antiparallel strong trace and the upper embeddability of a planar graph. In fact, an AO link of one component can naturally give the antiparallel strong trace of a planar graph. For example, Table 1 indicates that 4-pyramid, 6-pyramid, 3-prism and 5-prism have more than one antiparallel strong trace. For a 3-regular planar graph G, if the graph G allows the AO link diagram $D(G)$ of one or two components, G must be upper-embeddable since the graph G can be cellularly embedded on the orientable surface obtained by applying Seifert's algorithm to $D(G)$. Otherwise, G must not be upper embeddable. For a non 3-regular planar graph H, if H allows the AO link diagram of one or two components, the graph H must be upper embeddable. Otherwise, the graph H will be splitted into some 3-regular planar graphs by dividing each non 3-degree vertex into some 3-degree vertices [15]. The upper embeddability of H can be finally given by further checking whether the resulting 3-regular planar graphs allow the AO link diagram of one or two components using the software"AO Link". Hence the software"AO link" can be used to check whether a planar graph G is a upper-embeddable graph. As a result, all polyhedrons involved in Table 1 are upper-embeddable graphs.

***Acknowledgment*:** This work was financially supported by the National Natural Science Foundation of China [Grant No. 11501454], the Natural Science Foundation of Shandong Province [ZR2025MS224] and the Future Scholars Program of Shandong University.

Supplementary Materials: This includes the executable and input files of the software "*AO link* ", "*AO1 link* ", "*Dodecahedra*" and "*Icosahedra*", and all AO link diagrams counted in table 1 and their link components. Please download it from the following link: `https://pan.baidu.com/s/1cGk_HPYAh33BAUNitIsdBg?pwd=2682`

References

[1] R. Cromwell, *Knots and Links*, Cambridge Univ. Press, Cambridge, 2004.

[2] N. C. Seeman, H. Sleiman, DNA nanotechnology, *Nat. Rev. Mater* **3** (2018) #17068.

[3] Y. Li, C. Mao, Z. Deng, Supramolecular wireframe DNA polyhedra: assembly and applications, *Chin. J. Chem.* **35** (2017) 801–810.

[4] Y. Ke, J. Sharma, M. Liu, K. Jahn, Y. Liu, H. Yan, Scaffolded DNA origami of a DNA tetrahedron molecular container, *Nano Lett.* **9** (2009) 2445–2447.

[5] X. He, L. Dong, W. Wang, N. Lin, Y. Mi, Folding single-stranded DNA to form the smallest 3D DNA triangular prism, *Chem. Commun.* **49** (2013) 2906–2908.

[6] V. Kočar, J. S. Schreck, S. Čeru, H. Gradišar, N. Bašić, T. Pisanski, J. P. K. Doye, R. Jerala, Design principles for rapid folding of knotted DNA nanostructures, *Nat Commun.* **7** (2016) #10803.

[7] X. Qi, F. Zhang, Z. Su, S. Jiang, D. Han, B. Ding, Y. Liu, W. Chiu, P. Yin, H. Yan, Programming molecular topologies from single-stranded nucleic acids, *Nat. Commun.* **9** (2018) #4579.

[8] R. P. Goodman, R. M. Berry, A. J. Turberfield, The single-step synthesis of a DNA tetrahedron, *Chem. Commun.* **12** (2004) 1372–1373.

[9] T. Ye, A. E. Ribbe, C. Zhang, A. E. Ribbe, W. Jiang, C. Mao, Hierarchical self-assembly of DNA into symmetric supramolecular polyhedra, *Nature* **452** (7184) 198–201.

[10] N. C. Seeman, J. Chen, Synthesis from DNA of a molecule with the connectivity of a cube, *Nature* **350** (1991) 631–633.

[11] C. M. Erben, R. P. Goodman, A. J. Turberfield, A self-assembled DNA bipyramid, *J. Am. Chem. Soc.* **129** (207) 6992–6993.

[12] F. F. Andersen, B. Knudsen, C. L. P. Oliveira, R. F. Frøhlich, D. Krüger , J. Bungert, M. Agbandje-McKenna, R. McKenna, S. Juul, C. Veigaard, J. Koch, J. L. Rubinstein, B. Guldbrandtsen, M. S. Hede, G. Karlsson, A. H. Andersen, J. S. Pedersen, B. R. Knudsen, Assembly and structural analysis of a covalently closed nano-scale DNA cage, *Nucleic Acids Res.* **36** (2008) 1113–1119.

[13] Y. He, M. Su, P. Fang, C. Zhang, A. E. Ribbe, W. Jiang, C. Mao, On the chirality of self-assembled DNA octahedra, *Angew. Chem. Int.* **49** (2010) 748–751.

[14] N. Jonoska, M. Saito, Boundary components of thickened graphs, *Lect. Notes Comput. Sci.* **2340** (2002) 70–81.

[15] N. Jonoska, N. C. Seeman, G. Wu, On existence of reporter strands in DNA-based graph structures, *Theor. Comput. Sci.* **410** (2009) 1448–1460.

[16] G. Fijavž, T. Pisanski, J. Rus, Strong traces model of self-assembly polypeptide structures, *MATCH Commun. Math. Comput. Chem.* **71** (2014) 199–212.

[17] X. Cheng, Q. Deng, Y. Diao, Constructions of DNA and polypeptide cages based on plane graphs and odd crossing π-junctions, *Appl. Math. Comput.* **443** (2023) #127773.

[18] W. Qiu, Z. Wang, G. Hu, *The Chemistry and Mathematics of DNA Polyhedra*, Nova Sci. Pub., New York, 2010.

[19] S. Jablan, L. Radović, R. Sazdanović, Knots and links derived from prismatic graphs, *MATCH Commun. Math. Comput. Chem.* **66** (2010) 65–92.

[20] G. Hu, W. Qiu, A. Ceulemans, A new Euler's formula for DNA polyhedra, *PloS One* **6** (2011) #e26308.

[21] S. Liu, H. Zhang, Genera of the links derived from 2-connected plane graphs, *J. Knot Theory Ramif.* **21** (2012) #1250129.

[22] J. Duan, W. Qiu, Using dual polyhedral links models to predict characters of DNA polyhedra, *J. Mol. Struct.* **1051** (2013) 233–236.

[23] S. Liu, H. Zhang, Some invariants of polyhedral links, *MATCH Commun. Math. Comput. Chem.* **70** (2013) 383–400.

[24] X. Jin, A survey on several invariants of three types of polyhedral links, *MATCH Commun. Math. Comput. Chem.* **76** (2016) 569–594.

[25] J. Duan, L. Cui, Y. Wang, Rational design of DNA platonic polyhedra with the minimal components number from topological perspective, *Biochem. Bioph. Res. Co.* **523** (2020) 627–631.

[26] T. Deng, Z. Man, W. Wang, An assembling strategy for DNA cages with minimum strands, *Comput. Biol. Chem.* **93** (2021) #107507.

[27] S. Liu, L. Guo, H. Bai, J. Hao, A topological approach to assembling strands based DNA tetrahedra, *MATCH Commun, Math. Comput. Chem.* **81** (2019) 209–244.

[28] L. Guo, H. Bai, J. Hao, S. Liu, A topological approach to assembling strands-based DNA triangular prisms, *MATCH Commun. Math. Comput. Chem.* **82** (2019) 219–253.

[29] L. Guo, W. Sun, J. Hao, S. Liu, On the chirality of triangular prism links, *Match Commun. Math. Comput. Chem.* **82** (2019) 255–292.

[30] H. Lin, J. Li, S. Liu, The calculation of topological structures of strands-based DNA trigonal bipyramids, *J. Mol. Graph. Model.* **95** (2020) #107492.

[31] H. Lin, H. Zhang, S. Liu, On the HOMFLY polynomials of even trigonal bipyramid links, *J. Math. Chem.* **60** (2022) 542–554.

[32] Y. Lu, X. Guo, S. Liu, Topological structures of DNA octahedrons determined by the number of ssDNA strands, *J. Mol. Graph. Model.* **126** (2024) #108657.

[33] L. H. Kauffman, A Tutte polynomial for signed graphs, *Discr. Appl. Math.* **25** (1989) 105–127.

[34] J. A. Bondy, U. S. R. Murty, *Graph Theory With Applications*, Elsevier, North Holland, 1984.

[35] Y. Yin, M. Xiong, C. Liu, X. Zhang, Q. Ou, Cavity-engineered DNA nanocages: a multifunctional tool for biosensing, cargo delivery and materials assembly, *Sci. China Chem.* **68** (2025) 4746–4760.