Natural Computing
https://doi.org/10.1007/s11047-025-10053-6

A multipartite approach for the self-assembly of DNA graph structures

S. Bonvicini[1] · **M. M. Ferrari**[2]

Received: 18 December 2024 / Accepted: 19 August 2025

Abstract

We consider a graph theory problem motivated by the self-assembly of DNA graph structures using branched junction molecules with flexible arms (called 'tiles' in the combinatorial model). More precisely, we want to determine a set of tiles that realizes a target graph G using the minimum number of bond-edge types so that no graph with order smaller than $|V(G)|$ can be realized; the parameter of interest is denoted by $B_2(G)$. We present an approach that provides an upper bound for $B_2(G)$ using certain multipartite subgraphs of G. We provide some numerical conditions characterizing such multipartite graphs in terms of the degree of their vertices. Then, we apply our method to the graphs corresponding to the Platonic solids.

1 Introduction

In a 2023 paper written by Ellis-Monaghan and Jonoska (2023) on the occasion of the 40th anniversary of DNA nanotechnology, the authors highlight the existence of a new branch of mathematics, the so called 'DNA mathematics', which arises from DNA self-assembly as a method to build nanostructures. Seeman, with the paper Seeman (1982), is considered the one who initiated the field of DNA nanotechnology: his work influenced several areas of study such as computer science and mathematics (Ellingham and Ellis-Monaghan 2019; Ellis-Monaghan et al. 2014; Jonoska and McColm 2009; Morse et al. 2020). This paper falls in the category of 'DNA mathematics', since we study a graph theory problem arising from the self-assembly of DNA graph structures using branched junction molecules. These building blocks resemble the shape of stars having arms stemming from a central point; each arm is made of DNA strands and possesses a bonding site at its end called cohesive-end. A cohesive-end consists of an unpaired sequence of bases (A's, T's, C's, G's) so that branched junction molecules can attach together via arms having complementary cohesive-ends. See Figure 2 for a schematic illustration. Here, we consider molecules with flexible arms as those utilized in Wu et al. (2009), Sa-Ardyen et al. (2004) to assemble DNA graph structures; thus, we assume that each arm can freely move and bond with any other arm having complementary cohesive-end (unlike the case of rigid arms modeled in Ferrari et al. (2018), Jonoska et al. (2006)).

Motivated by the problem of efficiently building a graph structure using branched junction molecules, Ellis-Monaghan et al. (2014) focused on minimizing the number of distinct molecules and cohesive-end types required to build the desired construct under various laboratory conditions . As reviewed in Ellis-Monaghan et al. (2019), Ellis-Monaghan and Jonoska (2023), the self-assembly of branched junction molecules is modeled using graph theory where the molecules are represented as tiles (see Figure 2). Then, the above problem consists in computing two graph invariants, called the minimum number of tile types $T(G)$ and the minimum number of bond-edge types $B(G)$, for a given graph G. The exact value of these invariants is known for certain families of graphs, such as complete graphs, complete bipartite graphs, and cycles (Ellis-Monaghan et al. 2014; Bonvicini and Ferrari 2020), Platonic graphs, square lattice graphs, and triangular lattice graphs (Almodóvar et al. 2024; Almodovar et al. 2021), complete tripartite graphs and cocktail party graphs (Almodóvar et al. 2024), and other classes (Redmon et al. 2023; Griffin and Sorrells 2023;

✉ S. Bonvicini
simona.bonvicini@unimore.it

✉ M. M. Ferrari
margherita.ferrari@umanitoba.ca

[1] Dipartimento di Scienze Fisiche, Informatiche e Matematiche, Università di Modena e Reggio Emilia, via Campi 213/b, 41126 Modena, Italy

[2] Department of Mathematics, University of Manitoba, Winnipeg, MB R3T 2N2, Canada

Published online: 04 October 2025

Mattamira 2020; Almodóvar et al. 2019). Related computational complexity questions are discussed in Almodóvar et al. (2024) while algorithmic procedures to compute these parameters are presented in Ashworth et al. (2024). Using edge-colorings of graphs (Bonvicini and Ferrari 2020), the problem of determining $B(G)$ (respectively, $T(G)$) can be related to the one of finding an edge-coloring of type m (Hakimi and Kariv 1986) for G with the minimum number of colors (respectively, multi-palettes).

This work deals with the problem of determining a collection of tiles that realizes a target graph G but no graph with fewer vertices than G while minimizing the number of bond-edge types. As discussed in Sect. 4, the minimum number of bond-edge types for G under this constraint is denoted by $B_2(G)$. We introduce a method, which we call 'multipartite approach' as quoted in the title, that can be used to find an upper bound for the parameter $B_2(G)$ when G contains a prescribed ℓ-partite subgraph. This approach also shows a connection with integer linear programming, a tool used in Ashworth et al. (2024) as well. In fact, in Sect. 2 we define two problems, see Definitions 2.2 and 2.6, that describe the multipartite approach in terms of integer linear programming; this connection will be discussed later in Sect. 5. In Sect. 3, we provide conditions for a multipartite graph to satisfy Definition 2.2 or Definition 2.6. In Sect. 4, we recall the graph theoretical formalism to model the self-assembly of branched junction molecules. In Sect. 5, we use edge-colorings of graphs and our approach to find some upper bounds for the parameter $B_2(G)$. As an application of our method, we study $B_2(G)$ for the Platonic graphs in Sect. 6.

2 Preliminary definitions

In this section, we introduce the definitions that are at the core of the multipartite approach we present in this paper. We refer the reader to Bondy and Murty (2008) for standard graph theory notation and terminology.

Since multipartite graphs are the main component of our method, it is worth recalling their definition: a graph G is *multipartite* if we can partition its vertex set into ℓ subsets (or parts) $U_1, \ldots, U_\ell$, with $\ell \geq 2$, such that the vertices belonging to the same subset of the partition are never adjacent. In this case, we adopt the notation $G = (U_1, \ldots, U_\ell)$ and we say that G is an *ℓ-partite graph*. Note that 2-partite graphs are known as *bipartite* graphs; for ease of exposition, we will denote a bipartite graph with parts U_1, U_2 by $B(U_1, U_2)$. For $1 \leq i \leq \ell$, we denote by D_i the degree set of the vertices in U_i, that is, $D_i = \{\deg_G(u) : u \in U_i\}$. In the first definition, we introduce ℓ-partite graphs with a prescribed partition of the edge set.

Definition 2.1 Let $G = (U_1, \ldots, U_\ell)$ be a connected ℓ-partite graph, where $\ell \geq 3$, and let $I = [1\,..\,\ell]$ be the set of positive integers from 1 to ℓ. We say that $G = (U_1, \ldots, U_\ell)$ is the *edge disjoint union of the induced bipartite subgraphs* $B(U_i, U_{i+1})$ if its edge set can be partitioned into the edge sets $E(B(U_i, U_{i+1}))$ for $i \in I$ and where $E(B(U_\ell, U_1))$ may be the only empty set (subscripts are considered modulo ℓ).

For each subgraph $B(U_i, U_{i+1})$, we denote by $D_{i,i+1}$ and $D_{i+1,i}$ the degree sets of the vertices in $B(U_i, U_{i+1})$ belonging to U_i and U_{i+1}, respectively. Note that $D_{\ell,1} = D_{1,\ell} = \{0\}$ when $B(U_\ell, U_1)$ is edgeless.
For a degree set D_i consisting of a single element, say k, we will sometimes use the notation $d_i = k$. Similar notations will be used for the other degree sets considered in the paper.

We now define a system of linear diophantine equations and an integer linear programming problem associated with an ℓ-partite graph G satisfying Definition 2.1. The following definitions establish a connection between integer linear programming problems and DNA self-assembly that will be explained in detail in Remark 5.2.

For our purposes, we restrict the following definitions to the case in which each degree set D_i consists of a single (positive) element. Indeed, the purpose of this paper is to introduce our multipartite approach in a way that allows the reader to understand the spirit of the method. For this reason, we restrict the next definitions to the case where the degree sets consist of one or two elements. A formulation with degree sets consisting of more elements is possible but makes the definitions harder to read and, in our opinion, it does not allow the reader to grasp the general idea behind the constructions.

Definition 2.2 Let $G = (U_1, \ldots, U_\ell)$ be an ℓ-partite graph satisfying Definition 2.1 with $I = [1\,..\,\ell]$. Assume that the degree sets D_i, $D_{i,i-1}$, $D_{i,i+1}$ consist of the single element d_i, $d_{i,i-1}$, $d_{i,i+1}$, respectively, so that the equality $d_i = d_{i,i-1} + d_{i,i+1}$ is satisfied for $i \in I$.

For every $i \in I$, consider the polynomial $f_{i,i+1}(X_i, X_{i+1}) = d_{i,i+1}X_i - d_{i+1,i}X_{i+1}$ in the indeterminates X_i, X_{i+1} and with integer coefficients $d_{i,i+1}$, $d_{i+1,i}$. We associate G with the following *system of linear diophantine equations*:

$$f_{i,i+1}(X_i, X_{i+1}) = 0 \text{ for } i \in I.$$

We define the *integer linear programming problem associated with* G, denoted by *ILP*(G), as the following

integer linear programming problem in the ℓ indeterminates $X_1, \ldots, X_\ell$:

$$ILP(G)\begin{cases} \min(X_1 + \cdots + X_\ell) & \\ f_{i,i+1}(X_i, X_{i+1}) & = 0, \quad i \in I \\ X_1 + \cdots + X_\ell & > 0 \\ X_i \geq 0, & i \in I \\ X_i \in \mathbb{Z}, & i \in I \end{cases}$$

We say that the ℓ-partite graph G is a *solution* to the associated integer linear programming problem *ILP*(G) if its order is the minimum value of the objective function $f(X_1, \ldots, X_\ell) = X_1 + \cdots + X_\ell$ in the feasible region of *ILP*(G).

Remark 2.3 We note that Definition 2.1 and Definition 2.2 can also be given for a bipartite graph $B(U_1, U_2)$. In this case, such a graph is the edge disjoint union of the single subgraph $B(U_1, U_2)$. Thus, $D_1 = D_{1,2}$ and $D_2 = D_{2,1}$. For simplicity, in the following we will refer to Definition 2.1 and Definition 2.2 even in the case of bipartite graphs.

In the next two examples, we illustrate the above definitions. We consider the case of complete bipartite graphs $K_{m,n}$ with $\gcd(m, n) = 1$ in Example 2.4, and with $\gcd(m, n) > 1$ in Example 2.5; the results will be also used in the subsequent sections.

Example 2.4 We represent the complete bipartite graph $K_{m,n}$ with $\gcd(m, n) = 1$ by the "standard" partition of its vertex set, that is, $K_{m,n}$ with $\gcd(m, n) = 1$ is the bipartite graph $B(U_1, U_2)$ where $|U_1| = m$ and $|U_2| = n$. According to the Remark 2.3, $K_{m,n}$ is the edge disjoint union of $B(U_1, U_2)$ itself. The degree sets D_1, D_2 consist of the single elements n, m and coincide with the degree sets $D_{1,2}$, $D_{2,1}$, respectively. We have one linear diophantine equation associated with $K_{m,n}$: $n\,X_1 - m\,X_2 = 0$. The integer linear programming problem $ILP(K_{m,n})$ associated with $K_{m,n}$ is thus the following:

$$ILP(K_{m,n})\begin{cases} \min(X_1 + X_2) \\ n\,X_1 - m\,X_2 = 0 \\ X_1 + X_2 > 0 \\ X_1,\ X_2 \geq 0 \\ X_1,\ X_2 \in \mathbb{Z} \end{cases}$$

We can write the objective function of $ILP(K_{m,n})$ as the linear function $f(X_1) = \left(1 + \frac{n}{m}\right) X_1$, since the linear diophantine equation $n\,X_1 - m\,X_2 = 0$ yields $X_2 = \frac{n}{m}\,X_1$. It is easy to see that for coprime values of m and n, the minimum positive value of f is $f(m) = m + n$, that is, the complete bipartite graph $K_{m,n}$ with $\gcd(m, n) = 1$ is a solution to the associated $ILP(K_{m,n})$.

Example 2.5 Let $K_{m,n}$ be the complete bipartite graph with $\gcd(m, n) > 1$ and bipartition of the vertex set given by the pair (U, W) where $|U| = m$ and $|W| = n$. We select an arbitrary vertex $w \in W$ and represent $K_{m,n}$ as the 3-partite graph $G = (U_1, U_2, U_3)$ with $U_1 = \{w\}$, $U_2 = U$, $U_3 = W \setminus \{w\}$. Thus, G is the edge disjoint union of the induced bipartite subgraphs $B(\{w\}, U)$, $B(U, W \setminus \{w\})$ and $B(\{w\}, W \setminus \{w\})$. As for the degree sets, we have $d_1 = d_3 = m$, $d_2 = n$, $d_{1,2} = m$, $d_{2,1} = 1$, $d_{2,3} = n - 1$, and $d_{3,2} = m$. The integer linear programming problem *ILP*(G) associated with the 3-partite graph G is the thus following:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3) & \\ m\,X_1 - X_2 & = 0 \\ (n-1)\,X_2 - m\,X_3 & = 0 \\ X_1 + X_2 + X_3 & > 0 \\ X_i \geq 0, & 1 \leq i \leq 3 \\ X_i \in \mathbb{Z}, & 1 \leq i \leq 3 \end{cases}$$

It is easy to see that the system of linear diophantine equations associated with G yields $X_2 = m\,X_1$ and $X_3 = (n-1)\,X_1$. We can thus write the objective function of *ILP*(G) as the linear function $f(X_1) = (m + n)X_1$ and prove that the minimum positive value of f is $f(1) = m + n$, that is, the graph G, which is isomorphic to the complete bipartite graph $K_{m,n}$ with $\gcd(m, n) > 1$, is a solution to the associated *ILP*(G).

Note that the arguments above do not depend on the assumption that $\gcd(m, n) > 1$; in fact, they hold for every pair of positive integers m, n. We prefer to specify that m and n are not coprime because Example 2.5 will be used to treat the case of $K_{m,n}$ with $\gcd(m, n) > 1$ in Sect. 5.

For our purposes, in the next definition we slightly modify the assumptions in Definition 2.2 by assuming that exactly two of the degree sets $D_{i,i+1}$ contain two non-negative integers (note that at least one of these two values must be positive).

Definition 2.6 Let $G = (U_1, \ldots, U_\ell)$ be an ℓ-partite graph satisfying Definition 2.1 with $I = [1\,..\,\ell]$. Assume that there exist two degree sets, say $D_{t,t-1}$ and $D_{t,t+1}$ for $t \in I$, consisting of two elements: $D_{t,t-1} = \{d^1_{t,t-1}, d^2_{t,t-1}\}$ and

$D_{t,t+1} = \{d^1_{t,t+1}, d^2_{t,t+1}\}$. All other degree sets $D_{i,i-1}$, $D_{i,i+1}$ for $i \in I$, $i \neq t$, are assumed to contain the single element $d_{i,i-1}$, $d_{i,i+1}$, respectively. Further to the above, the equality $d_t = d^j_{t,t-1} + d^j_{t,t+1}$ holds for $j = 1, 2$ (while $d_i = d_{i,i-1} + d_{i,i+1}$ for $i \in I$, $i \neq t$).

For every $i \in I$, $i \neq t-1, t$, consider the polynomial $f_{i,i+1}(X_i, X_{i+1}) = d_{i,i+1}X_i - d_{i+1,i}X_{i+1}$ in the indeterminates X_i, X_{i+1} and with integer coefficients $d_{i,i+1}$, $d_{i+1,i}$.

For $i = t-1, t$, consider the polynomials $f_{t-1,t}(X_{t-1}, X_t, Y_t) = d_{t-1,t}X_{t-1} - d^1_{t,t-1}X_t - d^2_{t,t-1}Y_t$ and $f_{t,t+1}(X_t, Y_t, X_{t+1}) = d^1_{t,t+1}X_t + d^2_{t,t+1}Y_t - d_{t+1,t}\,X_{t+1}$ in the indeterminates X_{t-1}, X_t, Y_t and X_t, Y_t, X_{t+1}, with integer coefficients $d_{t-1,t}$, $d^1_{t,t-1}$, $d^2_{t,t-1}$, and $d^1_{t,t+1}$, $d^2_{t,t+1}$, $d_{t+1,t}$, respectively. We associate G with the following *system of linear diophantine equations*:

$$\begin{cases} f_{i,i+1}(X_i, X_{i+1}) = 0, & i \in I \\ & i \neq t-1, t \\ f_{t-1,t}(X_{t-1}, X_t, Y_t) = 0 \\ f_{t,t+1}(X_t, Y_t, X_{t+1}) = 0 \end{cases}$$

We define the *integer linear programming problem associated with G*, denoted by *ILP*(*G*), as the following integer linear programming problem in the $\ell + 1$ indeterminates $X_1, \ldots, X_\ell, Y_t$:

$$ILP(G)\begin{cases} \min(X_1 + \cdots + X_\ell + Y_t) \\ f_{i,i+1}(X_i, X_{i+1}) = 0, & i \in I \\ & i \neq t-1, t \\ f_{t-1,t}(X_{t-1}, X_t, Y_t) = 0 \\ f_{t,t+1}(X_t, Y_t, X_{t+1}) = 0 \\ X_1 + \cdots + X_\ell + Y_t > 0 \\ Y_t \geq 0,\ X_i \geq 0, & i \in I \\ Y_t,\ X_i \in \mathbb{Z}, & i \in I \end{cases}$$

We say that the ℓ-partite graph G is a *solution* to the associated integer linear programming problem *ILP*(*G*) if its order is the minimum value of the objective function $f(X_1, \ldots, X_\ell, Y_t) = X_1 + \cdots + X_\ell + Y_t$ in the feasible region of *ILP*(*G*).

In the next example, we illustrate Definition 2.6.

Example 2.7 Let $G = (U_1, U_2, U_3, U_4, U_5)$ be the 5-partite graph in Fig. 1 with $U_1 = \{u_1\}, U_3 = \{u_2, u_3\}, U_5 = \{u_4\}$, $U_2 = \{w_i : 1 \leq i \leq 4\}$ and $U_4 = \{w_i : 5 \leq i \leq 8\}$. It is easy to see that G is the edge disjoint union of the induced bipartite subgraphs $B(U_i, U_{i+1})$ for $1 \leq i \leq 5$. Moreover, vertices in $U_1 \cup U_3 \cup U_5$ have degree 4 and vertices in $U_2 \cup U_4$ have degree 2, that is, $d_1 = d_3 = d_5 = 4$ and $d_2 = d_4 = 2$. As for the degree sets $D_{i,i+1}$, we have $d_{1,2} = 4$, $d_{2,1} = 1$, $d_{2,3} = 1$, $D_{3,2} = \{1, 3\}$, $D_{3,4} = \{1, 3\}$, $d_{4,3} = 1$, $d_{4,5} = 1$, $d_{5,4} = 4$.

The two degree sets $D_{t,t-1}$, $D_{t,t+1}$ consisting of exactly two elements are $D_{3,2} = \{1, 3\}$, $D_{3,4} = \{1, 3\}$; so $t = 3$ and the equality $d_t = d^j_{t,t-1} + d^j_{t,t+1}$ in Definition 2.6 holds for $j = 1, 2$, since $d_t = 4$, $d^1_{t,t-1} = 1$, $d^1_{t,t+1} = 3$ and $d^2_{t,t-1} = 3$, $d^2_{t,t+1} = 1$. The integer linear programming problem *ILP*(*G*) associated with the 5-partite graph G is thus the following:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3 + X_4 + X_5 + Y_3) \\ 4X_1 - X_2 = 0 \\ X_2 - X_3 - 3\,Y_3 = 0 \\ 3\,X_3 + Y_3 - X_4 = 0 \\ X_4 - 4\,X_5 = 0 \\ X_1 + X_2 + X_3 + X_4 + X_5 + Y_3 > 0 \\ Y_3 \geq 0,\ X_i \geq 0, & 1 \leq i \leq 5 \\ Y_3,\ X_i \in \mathbb{Z}, & 1 \leq i \leq 5 \end{cases}$$

The first and the last linear diophantine equation yield $X_2 = 4\,X_1$ and $X_4 = 4\,X_5$, respectively, that replaced in the second and third equations give $X_3 + 3\,Y_3 = 4\,X_1$

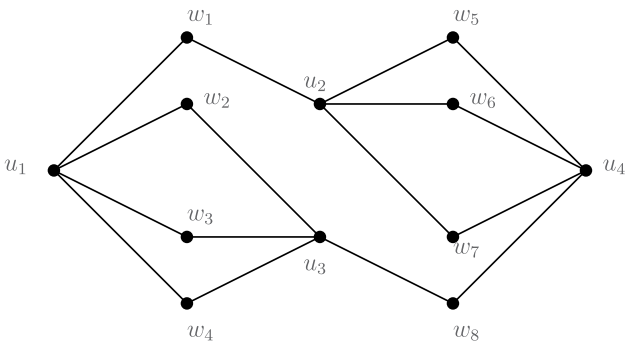

Fig. 1 The 5-partite graph $G = (U_1, U_2, U_3, U_4, U_5)$ in Example 2.7 with $U_1 = \{u_1\}, U_3 = \{u_2, u_3\}, U_5 = \{u_4\}, U_2 = \{w_i : 1 \leq i \leq 4\}$ and $U_4 = \{w_i : 5 \leq i \leq 8\}$. The graph G satisfies the assumptions in Proposition 3.1 and is isomorphic to the bipartite graph G' defined by the bipartition (U, W) with $U = \{u_i : 1 \leq i \leq 4\}$ and $W = \{w_i : 1 \leq i \leq 8\}$

and $3X_3 + Y_3 = 4X_5$; the sum of these last two equations yields $X_3 + Y_3 = X_1 + X_5$. The objective function $f(X_1, X_2, X_3, X_4, X_5, Y_3) = \sum_{i=1}^{5} X_i + Y_3$ can thus be written as $f(X_1, X_5) = 6(X_1 + X_5)$, and its minimum positive value is $f(1, 1) = 12$. It follows that G is a solution to the associated $ILP(G)$.

Remark 2.8 Let $G = (U_1, \ldots, U_\ell)$ be an ℓ-partite graph satisfying Definition 2.2. Suppose that the degree sets $D_{i,i-1}$, $D_{i,i+1}$ consist of the single element $d_{i,i-1}$, $d_{i,i+1}$, respectively. Then, the number of edges in $B(U_i, U_{i+1})$ is equal to $d_{i,i+1} \cdot |V(U_i)|$, or equivalently, $d_{i+1,i} \cdot |V(U_{i+1})|$. Hence, $X_i^* = |V(U_i)|$ and $X_{i+1}^* = |V(U_{i+1})|$ satisfy $f_{i,i+1}(X_i, X_{i+1}) = 0$ for $i \in I$. Similar observations hold for an ℓ-partite graph as in Definition 2.6.

3 Some numerical conditions

In this section, we give some sufficient conditions for an ℓ -partite graph to be a solution to the associated integer linear programming problem in Definition 2.2 or Definition 2.6. For ease of exposition, we restrict the number ℓ of parts to $\ell \leq 5$.

The following numerical conditions arise from the cases we came across in studying the self-assembly of graphs having a spanning ℓ-partite subgraph and, not coincidentally, turn out to be useful in proving that an ℓ-partite graph is a solution to a suitable integer linear programming problem. Examples 2.4, 2.5, and 2.7 illustrate this idea for some bipartite graphs $G' = (U, W)$ where the size of U and W, together with the degree of the vertices, allow us to split the vertex set of G' into a convenient number ℓ of parts so that G' can be represented as an ℓ-partite graph G that satisfies the assumptions in Definition 2.2 or 2.6 and is a solution to the associated integer linear programming problem $ILP(G)$.

The first numerical condition we prove is a generalization of Example 2.7.

Proposition 3.1 *Let $G = (U_1, U_2, U_3, U_4, U_5)$ be a 5-partite graph satisfying the assumptions in Definition 2.6 with $d_1 = d_3 = d_5 = d$, $d_2 = d_4 = d'$ and $t = 3$. Assume that the last induced bipartite subgraph $B(U_5, U_1)$ is edgeless.*

1. *If $(d + d')$ is divisible by $d_{2,1} \cdot d_{4,5}$ and the inequality $[(d + d')(d_{2,1} + d_{4,5})]/(d_{2,1} \cdot d_{4,5}) \geq |V(G)|$ holds, then G is a solution to the associated integer linear programming problem ILP(G) in Definition 2.6.*
2. *If $(d + d')$ is not divisible by $d_{2,1} \cdot d_{4,5}$, let k be the smallest positive integer such that $(d + d')\gcd(d_{2,1}, d_{4,5})k \equiv 0 \pmod{d_{2,1} \cdot d_{4,5}}$. If the inequality $[(d + d')\gcd(d_{2,1}, d_{4,5})k]/(d_{2,1} \cdot d_{4,5}) \geq |V(G)|$ holds, then G is a solution to the associated integer linear programming problem ILP(G) in Definition 2.6.*

Proof By the assumptions, the integer linear programming problem associated with G in Definition 2.6 is the following:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3 + X_4 + X_5 + Y_3) & \\ d\,X_1 - d_{2,1}\,X_2 & = 0 \\ d_{2,3}\,X_2 - d_{3,2}^1\,X_3 - d_{3,2}^2\,Y_3 & = 0 \\ d_{3,4}^1\,X_3 + d_{3,4}^2\,Y_3 - d_{4,3}\,X_4 & = 0 \\ d_{4,5}\,X_4 - d\,X_5 & = 0 \\ X_1 + X_2 + X_3 + X_4 + X_5 + Y_3 & > 0 \\ Y_3 \geq 0,\ X_i \geq 0, & 1 \leq i \leq 5 \\ Y_3,\ X_i \in \mathbb{Z}, & 1 \leq i \leq 5 \end{cases}$$

The first and the last linear diophantine equation yield $d_{2,1}\,X_2 = d\,X_1$ and $d_{4,5}\,X_4 = d\,X_5$, respectively. From the second and the third equation, we obtain $d\,(X_3 + Y_3) = d_{2,3}\,X_2 + d_{4,3}\,X_4$ since $d_t = d_3 = d$ and $d_t = d_{t,t-1}^j + d_{t,t+1}^j$ holds for $j = 1, 2$. Multiplying both sides of this last equality by $d_{2,1} \cdot d_{4,5}$ and keeping in mind that $d_{2,1}\,X_2 = d\,X_1$, $d_{4,5}\,X_4 = d\,X_5$, we have that $d_{2,1} \cdot d_{4,5}\,(X_3 + Y_3) = d_{2,3} \cdot d_{4,5}\,X_1 + d_{4,3} \cdot d_{2,1}\,X_5$. We write this last equality as $d_{2,1} \cdot d_{4,5}\,(X_3 + Y_3) = (d' - d_{2,1}) \cdot d_{4,5}\,X_1 + (d' - d_{4,5}) \cdot d_{2,1}\,X_5$, since $d_2 = d_{2,1} + d_{2,3}$, $d_4 = d_{4,3} + d_{4,5}$ and $d_2 = d_4 = d'$. By the above relations and an easy computation, we find that $d_{2,1} \cdot d_{4,5} \cdot (\sum_{i=1}^{5} X_i + Y_3) = (d + d')(d_{4,5}\,X_1 + d_{2,1}\,X_5)$, whence $(d + d')(d_{4,5}\,X_1 + d_{2,1}\,X_5) \equiv 0 \pmod{d_{2,1} \cdot d_{4,5}}$. We now consider the following cases:

1. If $(d + d')$ is divisible by $d_{2,1} \cdot d_{4,5}$, then $\sum_{i=1}^{5} X_i + Y_3 = [(d + d')(d_{4,5}\,X_1 + d_{2,1}\,X_5)]/(d_{2,1} \cdot d_{4,5}) \geq [(d + d')(d_{4,5} + d_{2,1})]/(d_{2,1} \cdot d_{4,5})$, since we cannot have $\{X_1, X_5\} = \{0, 1\}$ in any solution to $ILP(G)$. It thus follows from the assumptions that $\sum_{i=1}^{5} X_i + Y_3 \geq |V(G)|$; whence, G is a solution to the associated integer linear programming problem in Definition 2.6.
2. If $(d + d')$ is not divisible by $d_{2,1} \cdot d_{4,5}$, then $(d + d')(d_{4,5}\,X_1 + d_{2,1}\,X_5) \equiv 0 \pmod{d_{2,1} \cdot d_{4,5}}$ implies that $(d_{4,5}\,X_1 + d_{2,1}\,X_5)$ is a positive value, say c, such that $(d + d')c$ is divisible by $d_{2,1} \cdot d_{4,5}$. By setting $(d_{4,5}\,X_1 + d_{2,1}\,X_5) = c$, we obtain a linear diophantine equation that admits a solution if and only if c is divisible by $\gcd(d_{2,1}, d_{4,5})$. Therefore, we can set $(d_{4,5}\,X_1 + d_{2,1}\,X_5) = \gcd(d_{2,1}, d_{4,5})h$, where h is a positive integer, and obtain that $d_{2,1} \cdot d_{4,5} \cdot (\sum_{i=1}^{5} X_i + Y_3) =$

$(d + d')\gcd(d_{2,1}, d_{4,5})h \geq (d + d')\gcd(d_{2,1}, d_{4,5})k$. By the assumptions, we get the inequality $\sum_{i=1}^{5} X_i + Y_3 \geq [(d + d')\gcd(d_{2,1}, d_{4,5})k]/(d_{2,1} \cdot d_{4,5}) \geq |V(G)|$; whence, G is a solution to the associated integer linear programming problem in Definition 2.6. □

Note that Example 2.7 shows how to apply Proposition 3.1 to bipartite graphs. The bipartite graph in the example is the bipartite graph G', isomorphic to the graph G of the example, that is defined by the bipartition (U, W) with $U = \{u_i : 1 \leq i \leq 4\}$, $W = \{w_i : 1 \leq i \leq 8\}$, and vertices in U, W of degree 4, 2, respectively. Proposition 3.1 can be considered as a generalization of Example 2.7.

In the next proposition, we consider the case of a 3-partite graph G that, unlike the 5-partite graph in the above result, has the last induced subgraph $B(U_\ell, U_1)$ not edgeless.

Proposition 3.2 *Let $G = (U_1, U_2, U_3)$ be a 3-partite graph satisfying the assumptions in Definition 2.6 with $d_1 = d$, $d_2 = d_3 = d'$, $t = 1$, $d^1_{1,2} = d^2_{1,3} = d$, and $d_{2,1} = d_{3,1}$.*

If $\gcd(d, 2\,d_{2,1}) = 1$ and the inequality $2(d + d_{2,1}) \geq |V(G)|$ holds, then G is a solution to the associated integer linear programming problem ILP(G) in Definition 2.6.

Proof The integer linear programming problem associated with G in Definition 2.6 is the following:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3 + Y_1) & \\ d^1_{1,2}\,X_1 + d^2_{1,2}\,Y_1 - d_{2,1}\,X_2 & = 0 \\ d_{2,3}\,X_2 - d_{3,2}\,X_3 & = 0 \\ d^1_{1,3}\,X_1 + d^2_{1,3}\,Y_1 - d_{3,1}\,X_3 & = 0 \\ X_1 + X_2 + X_3 + Y_1 & > 0 \\ Y_1 \geq 0,\ X_i \geq 0, & 1 \leq i \leq 3 \\ Y_1,\ X_i \in \mathbb{Z}, & 1 \leq i \leq 3 \end{cases}$$

The assumption $d_{2,1} = d_{3,1}$ implies that $d_{2,3} = d_{3,2}$ since the equalities $d_{2,3} = (d_2 - d_{2,1}) = (d' - d_{3,1}) = d_{3,2}$ hold. Bearing in mind that $d^1_{1,2} = d^2_{1,3} = d$, so that $d^2_{1,2} = d^1_{1,3} = 0$, we can write the above *ILP*(G) as follows:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3 + Y_1) & \\ d\,X_1 - d_{2,1}\,X_2 & = 0 \\ d_{2,3}\,X_2 - d_{2,3}\,X_3 & = 0 \\ d\,Y_1 - d_{2,1}\,X_3 & = 0 \\ X_1 + X_2 + X_3 + Y_1 & > 0 \\ Y_1 \geq 0,\ X_i \geq 0, & 1 \leq i \leq 3 \\ Y_1,\ X_i \in \mathbb{Z}, & 1 \leq i \leq 3 \end{cases}$$

Since the second linear diophantine equation yields $X_2 = X_3$, the first and the last equations yield $d\,(X_1 + Y_1) = 2\,d_{2,1}\,X_2$. If $\gcd(d, 2\,d_{2,1}) = 1$, then the equation $d\,(X_1 + Y_1) = 2\,d_{2,1}\,X_2$ is satisfied if and only if $X_1 + Y_1 = 2\,d_{2,1}\lambda$ and $X_2 = d\lambda$ for a suitable positive integer λ. We thus have $X_1 + X_2 + X_3 + Y_1 = 2\,d_{2,1}\lambda + 2d\lambda = 2\lambda(d + d_{2,1})$. It follows from the assumptions that $2\lambda(d + d_{2,1}) \geq |V(G)|$; whence, G is a solution to the associated integer linear programming problem *ILP*(G) in Definition 2.6. □

The last numerical condition deals with the case of regular 4-partite graphs.

Proposition 3.3 *Let $G = (U_1, U_2, U_3, U_4)$ be a 4-partite graph satisfying the assumptions in Definition 2.2 with $d_1 = d_2 = d_3 = d_4 = d$, $d_{1,2} = d_{4,3} = d$, $d_{2,3} = d_{3,2}$ and $d_{2,1} = d_{3,4}$.*

If $\gcd(d, d_{2,1}) = 1$ and the inequality $2(d + d_{2,1}) \geq |V(G)|$ holds, then G is a solution to the associated integer linear programming problem ILP(G) in Definition 2.2.

Proof Notice that the assumption $d_{1,2} = d_{4,3} = d$ implies that the last induced bipartite subgraph $B(U_4, U_1)$ is edgeless. By taking into account the other assumptions on the degrees, we can write the integer linear programming problem associated with G in Definition 2.2 as follows:

$$ILP(G)\begin{cases} \min(X_1 + X_2 + X_3 + X_4) & \\ d\,X_1 - d_{2,1}\,X_2 & = 0 \\ d_{2,3}\,X_2 - d_{2,3}\,X_3 & = 0 \\ d_{2,1}\,X_3 - d\,X_4 & = 0 \\ X_1 + X_2 + X_3 + X_4 & > 0 \\ X_i \geq 0, & 1 \leq i \leq 4 \\ X_i \in \mathbb{Z}, & 1 \leq i \leq 4 \end{cases}$$

The second linear diophantine equation yields $X_2 = X_3$, and so $d\,X_1 = d\,X_4 = d_{2,1}\,X_2$. If $\gcd(d, d_{2,1}) = 1$, then the equations $d\,X_1 = d\,X_4 = d_{2,1}\,X_2$ are satisfied if and only if $X_1 = X_4 = d_{2,1}\lambda$ and $X_2 = X_3 = d\lambda$ for a suitable positive integer λ. We thus have that $X_1 + X_2 + X_3 + X_4 = 2\lambda(d + d_{2,1})$. It follows from the assumptions that $2\lambda(d + d_{2,1}) \geq |V(G)|$; whence, G is a solution to the associated integer linear programming problem $ILP(G)$ in Definition 2.2. □

In the next example, we apply Proposition 3.2 and Proposition 3.3 to a bipartite graph $G' = (U, W)$ where $|U| = |W| = n$, $n \geq 3$, and all vertices have degree $d = n - 1$.

Example 3.4 Let $G' = (U, W)$ be a bipartite graph with $|U| = |W| = n$, $n \geq 3$, and all vertices of degree $d = n - 1$. Let $u \in U$ and $w \in W$ be non-adjacent vertices, and let $N(u)$, $N(w)$ be the neighborhood of u, w in G', respectively.

For even values of n, we represent G' as the 3-partite graph $G = (U_1, U_2, U_3)$ where $U_1 = \{u, w\}$, $U_2 = N(u)$, $U_3 = N(w)$. It is easy to see that G satisfies the assumptions in Proposition 3.2 with $d_1 = d_2 = d_3 = n - 1$, $d^1_{1,2} = d^2_{1,3} = n - 1$ and $d_{2,1} = d_{3,1} = 1$. Moreover, the conditions $\gcd(d, 2d_{2,1}) = 1$ and $2(d + d_{2,1}) \geq |V(G)|$ are also satisfied. Thus, it follows from Proposition 3.2 that G is a solution to the associated integer linear programming problem $ILP(G)$ in Definition 2.6, provided that n is even.

For odd values of n, we represent G' as the 4-partite graph $G = (U_1, U_2, U_3, U_4)$ where $U_1 = \{u\}$, $U_2 = N(u)$, $U_3 = N(w)$, $U_4 = \{w\}$. It is easy to see that G satisfies the assumptions in Proposition 3.3 with $d_1 = d_2 = d_3 = d_4 = n - 1$, $d_{1,2} = d_{4,3} = n - 1$, $d_{2,3} = d_{3,2} = n - 2$ and $d_{2,1} = d_{3,4} = 1$. Moreover, the conditions $\gcd(d, d_{2,1}) = 1$ and $2(d + d_{2,1}) \geq |V(G)|$ are also satisfied. Thus, it follows from Proposition 3.3 that G is a solution to the associated integer linear programming problem $ILP(G)$ in Definition 2.2, provided that n is odd.

4 Graph theoretical formalism for self-assembly

In this section, we describe the graph theoretical formalism used to model the self-assembly of branched junction molecules. The same notions can also be found in Ellis-Monaghan et al. (2014), Ellis-Monaghan et al. (2019), Jonoska et al. (2006), Jonoska et al. (2011). We consider graphs that are connected and may have loops or multiple edges. We write $e = \{u, v\}$ to indicate that the edge e is incident with the (not necessarily distinct) vertices u, v; the half-edges incident with u and v will be denoted by (u, e) and (v, e), respectively. As such, every edge $e = \{u, v\}$ can be thought as the join of the half-edges incident with u and v, respectively. The degree of a vertex is the number of half-edges incident with it.

Definition 4.1 A branched junction molecule with k flexible arms is modeled as a vertex with k labeled half-edges, called a *tile*. The labels represent the *cohesive-end types* and belong to a set $\{a, \hat{a} : a \in \Sigma\}$, where Σ is a finite set of symbols; complementary cohesive-end types are denoted by "hatted" and "unhatted" copies of the same symbol, with the convention that $\hat{\hat{a}} = a$ and $a \neq \hat{a}$ for every $a \in \Sigma$. A tile is denoted by the multiset consisting of the labels on its half-edges. We say that two tiles are of the same *tile type* if they are represented by the same multiset.

We will use the power notation, say a^m, to indicate that the label a appears m times in a multiset. Thus, for instance, the multiset $\{a^m, b\}$ consists of the symbol a appearing m times, together with the symbol b appearing exactly once. To further simplify the notation, the multiset $\{a^m, b\}$ will be denoted by $\{a^m b\}$.

Definition 4.2 Let t and t' be tiles (not necessarily of different tile type) corresponding to two branched junction molecules. A bond between the two molecules through arms with complementary cohesive-ends is modeled by joining the tiles t and t' by an edge that is the join of a half-edge of t labeled by, say a, with a half-edge of t' labeled by $\hat{a}$. The edge thus obtained is called a *bond-edge of type a*.

Figure 2 summarizes the above definitions.

The following definition is given by using the notion of assembly design in Ellis-Monaghan et al. (2019) and by using edge-colorings and orientations of graphs in Ferrari et al. (2023).

Definition 4.3 Let G be a graph allowing loops and multiple edges. A set of tile types $\mathcal{T}$, called a *pot*, *realizes* G if we can assign a tile type in $\mathcal{T}$ to each vertex v of G and its incident half-edges in such a way that:

(i) there is a bijection between the half-edges of v and the cohesive-end types of the corresponding tile type;
(ii) if $e = \{u, v\}$ is an edge of G, then the half-edges (u, e) and (v, e) are assigned complementary cohesive-end types, that is, the edge e corresponds to a bond-edge type.

As an example, the pot $\mathcal{T} = \{\{a^2\}, \{\hat{a}b\}, \{\hat{b}^2\}\}$ realizes a 4-cycle as well as an 8-cycle; see Figure 3. A DNA complex assembled from branched junction molecules is called

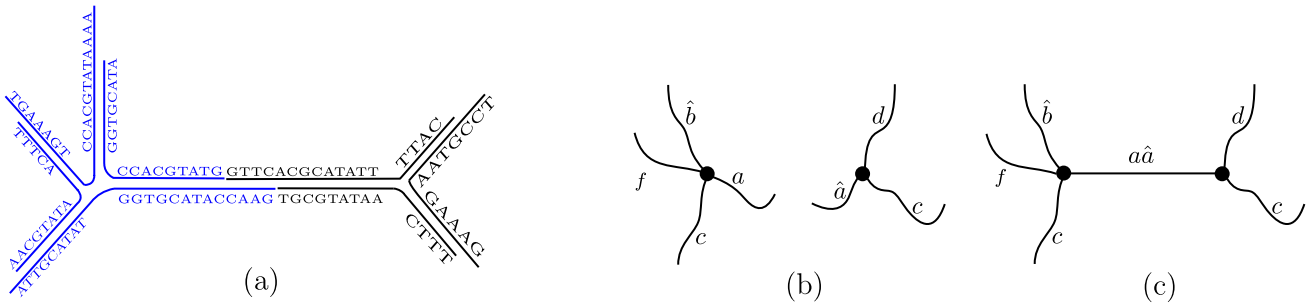

Fig. 2 (**a**) A schematic description of the join of two branched junction molecules having flexible arms with complementary cohesive-ends. (**b**) The two molecules are modeled by 4- and 3-valent vertices with labeled half-edges: the half-edges correspond to the arms of the molecules, and the labels correspond to the cohesive-end types. (**c**) From the join of two half-edges with the complementary labels a, $\hat{a}$, we obtain a bond-edge of type a. The figure is adapted from Bonvicini and Ferrari (2020)

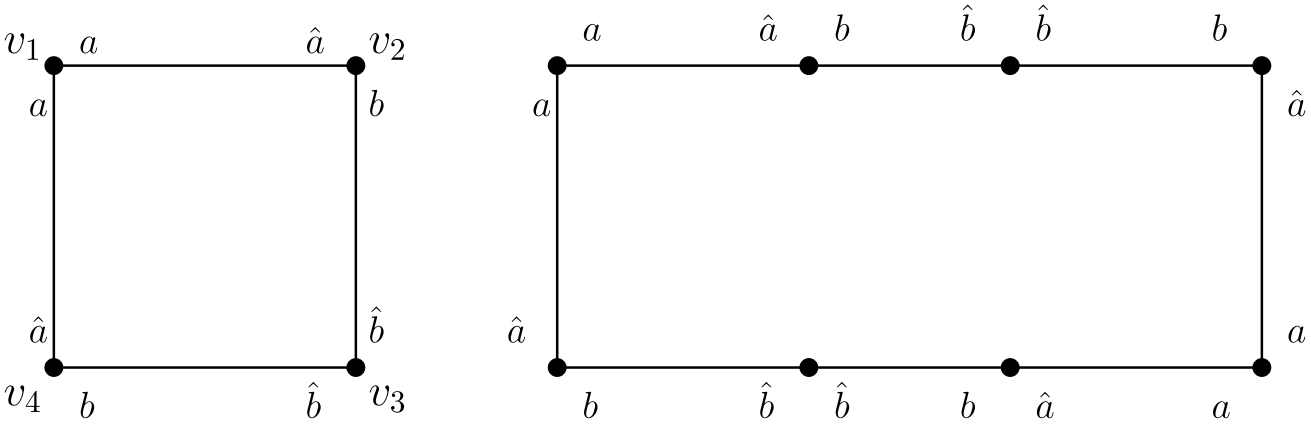

Fig. 3 The pot $\mathcal{T} = \{\{a^2\}, \{\hat{a}b\}, \{\hat{b}^2\}\}$ realizes both cycles in the figure. We can assign a tile type $t \in \mathcal{T}$ to every vertex v in such a way that there is a bijection between the half-edges incident with v and the cohesive-end types of t (for example, the half-edges incident with v_1 are all assigned with the label a). Observe also that each edge corresponds to a bond-edge type (for example, the vertices v_1, v_2 are assigned the tile types $\{a^2\}$, $\{\hat{a}b\}$, respectively, so that the edge $e = \{v_1, v_2\}$ receives both complementary cohesive-end types a and $\hat{a}$; thus, e is a bond-edge of type a)

complete if it has no molecule with unmatched cohesive-ends; the corresponding graph has no unmatched half-edges and, by analogy, is called a *complete complex*.

As shown in Fig. 3, a pot that realizes a graph G may realize other graphs as well. Minimizing the number of distinct tile types and/or of distinct bond-edge types may serve to improve the yield rates to self-assemble a target DNA complex with the underlying structure of G by reducing the incidental creation of undesired constructs. As such, the problem of determining a pot with the minimum number of tile types or of bond-edge types needed to realize a target graph G, is considered under three different conditions (Ellis-Monaghan et al. 2014):

- Scenario 1. The pot realizing G may realize graphs of order smaller than $|V(G)|$.
- Scenario 2. The pot realizes the graph G and no graph of order smaller than $|V(G)|$.
- Scenario 3. The pot realizes the graph G and no graph of order smaller than $|V(G)|$ or having the same order as G but not isomorphic to G.

For a target graph G, $T_i(G)$ (respectively, $B_i(G)$) denotes the minimum number of tile types (respectively, bond-edge types) needed to realize G in Scenario i, for $i = 1, 2, 3$ (Ellis-Monaghan et al. 2014). Definitions of these parameters using the notion of output of a pot can be found in Ellis-Monaghan et al. (2019), Ferrari et al. (2023). We remark that the problem of computing these two graph invariants is considered for graphs that are complete complexes. In the following, we will always assume that a graph G has no unmatched half-edges, unless otherwise specified.

In Bonvicini and Ferrari (2020), we addressed the problem of determining $T_i(G)$ and $B_i(G)$ using edge-colorings of graphs. By this colored approach, a bond-edge of a given type, say a, is nothing but an edge of color a. A pot $\mathcal{T}$ realizing a graph G gives rise to an edge-coloring of G, and also the converse holds. For example, the pot

$\mathcal{T} = \{\{a^2\}, \{\hat{a}b\}, \{\hat{b}^2\}\}$ realizing the 4-cycle G in Figure 3 is associated with the edge-coloring f of G that colors the edges $\{v_1, v_2\}$, $\{v_1, v_4\}$ with color a, and the remaining edges with color b. Vice versa, as shown in Figure 4, both pots $\mathcal{T} = \{\{a^2\}, \{\hat{a}b\}, \{\hat{b}^2\}\}$ and $\mathcal{T}' = \{\{a\hat{a}\}, \{a\hat{b}\}, \{\hat{a}b\}, \{b\hat{b}\}\}$ are associated with the same edge-coloring f. The figure also points out that the edge-colorings we consider are not necessarily proper. Formal definitions for the notions of tile etc. based on this colored approach can be found in Bonvicini and Ferrari (2020), Ferrari et al. (2023). Here, we prefer to introduce this set-up in a more informal way for ease of exposition.

It is thus easy to see that $B_1(G) = 1$ for every graph G, since there are no constraints on coloring the edges of G (the same result can also be found in Ellis-Monaghan et al. (2014)). For the parameters $B_2(G)$ and $B_3(G)$, the exact value for an arbitrary graph G is not known, and even lower and upper bounds may be challenging to find.

5 Bounds for $B_2(G)$

In this section, we construct pots $\mathcal{T}$ realizing the ℓ-partite graphs considered in Section 2, and will use them to find upper bounds for the parameter $B_2(G)$ of a given graph G. As in Bonvicini and Ferrari (2020), we construct pots starting from edge colorings of graphs. More specifically, given an edge coloring of a graph G, we split every edge of color, say a, into two half-edges: one will be labeled with a and the other with $\hat{a}$; the two labeled half-edges will thus contribute to define a tile type containing a, $\hat{a}$ or both labels. For the graphs discussed in this work, we use the following strategy.

Strategy for multipartite graphs:
Let G be an ℓ-partite graph as in Definition 2.1.

- Define the edge coloring that colors by a_i the edges of the subgraph $B(U_i, U_{i+1})$ for $1 \le i < \ell$, and colors by a_1 the edges of $B(U_\ell, U_1)$ if its edge set is non-empty. Notice that $\ell - 1$ colors have been used.
- Split every edge of color a_i into two half-edges labeled with a_i or $\hat{a}_i$ according to whether they are incident to the vertices in U_i or U_{i+1}, respectively, for $1 \le i < \ell$. If the edge set of $B(U_\ell, U_1)$ is non-empty, label with a_i (respectively, $\hat{a}_i$) the half-edges incident to the vertices in U_1 (respectively, U_ℓ).
- If the ℓ-partite graph G satisfies the assumptions in Definition 2.2, we then obtain a pot $\mathcal{T}$ realizing G containing the following tile types:
  - The vertices in U_1 have tile type $t_1 = \{a_1^{d_1}\}$;
  - The vertices in U_i, with $2 \le i \le \ell - 1$, have tile type $t_i = \{\hat{a}_{i-1}^{\lambda} a_i^{\mu}\}$ where $\lambda = d_{i,i-1}$ and $\mu = d_{i,i+1}$;
  - The vertices in U_ℓ have tile type $t_\ell = \{\hat{a}_1^{d_{\ell,1}} \hat{a}_{\ell-1}^{d_{\ell,\ell-1}}\}$, which becomes $t_\ell = \{\hat{a}_{\ell-1}^{d_{\ell,\ell-1}}\}$ if $d_{\ell,1} = 0$, that is, if $B(U_\ell, U_1)$ contains no edges.
- If the ℓ-partite graph G satisfies the assumptions in Definition 2.6, the definition of $\mathcal{T}$ is the same as above, except for the vertices in U_t that have tile type $\{\hat{a}_{t-1}^{\lambda} a_t^{\mu}\}$ with $(\lambda, \mu) = (d^1_{t,t-1}, d^1_{t,t+1})$ or $(\lambda, \mu) = (d^2_{t,t-1}, d^2_{t,t+1})$.

Example 5.1 The pot $\mathcal{T}$ realizing the complete bipartite graph $K_{m,n}$ is $\mathcal{T} = \{\{a_1^n\}, \{\hat{a}_1^m\}\}$ if $\gcd(m, n) = 1$, and $\mathcal{T} = \{\{a_1^m\}, \{\hat{a}_1 a_2^{n-1}\}, \{\hat{a}_2^m\}\}$ if $\gcd(m, n) > 1$ (see Example 2.4 and Example 2.5). Figure 5 shows the construction of the pot $\mathcal{T}$ realizing the complete bipartite graph $K_{2,3}$ in case (a), and $K_{3,3}$ in case (b). Figure 6 is about a graph satisfying the assumptions in Definition 2.6: it shows the construction of the pot $\mathcal{T} = \{\{a_1^4\}, \{\hat{a}_1 a_2\}, \{\hat{a}_2 a_3^3\}, \{\hat{a}_2^3 a_3\}, \{\hat{a}_3 a_4\}, \{\hat{a}_4^4\}\}$ realizing the graph G in Example 2.7.

Remark 5.2 Let $G = (U_1, \ldots, U_\ell)$ be an ℓ-partite graph as in Definition 2.2. As noted in Remark 2.8, $X_i^* = |V(U_i)|$ and $X_{i+1}^* = |V(U_{i+1})|$ is a solution to the linear diophantine

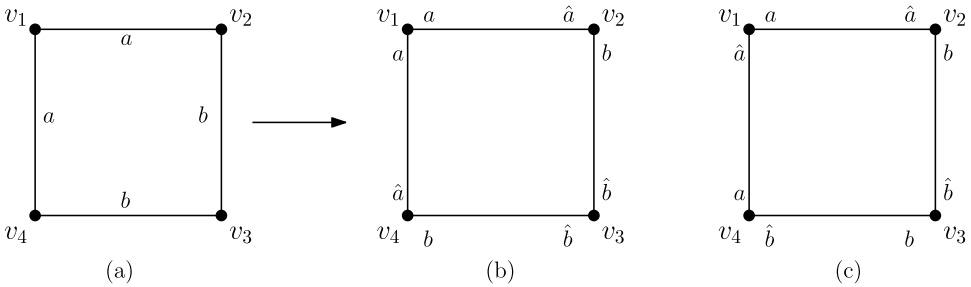

Fig. 4 (**a**) An edge-coloring f of a 4-cycle G that colors the edges $\{v_1, v_2\}$, $\{v_1, v_4\}$ with color a, and the remaining edges with color b. From every edge of a given color, say a, we obtain a labeling of the two corresponding half-edges with the symbols a and $\hat{a}$. From a labeling of the half-edges of G, we can then "read off" a pot realizing the graph G. A pot associated with an edge-coloring is not necessarily uniquely determined. In fact, the pots $\mathcal{T} = \{\{a^2\}, \{\hat{a}b\}, \{\hat{b}^2\}\}$ and $\mathcal{T}' = \{\{a\hat{a}\}, \{a\hat{b}\}, \{\hat{a}b\}, \{b\hat{b}\}\}$ are associated with the edge-coloring f in case (**b**) and case (**c**), respectively

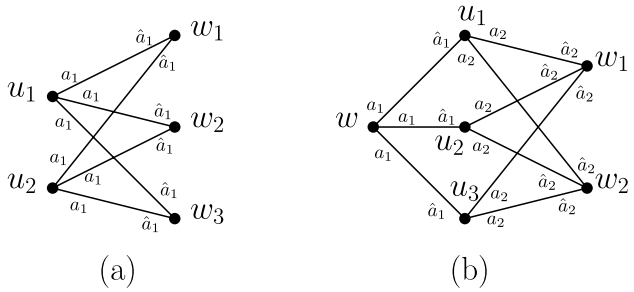

Fig. 5 (**a**) The complete bipartite graph $K_{2,3}$ in its "standard" vertex partition $U_1 = \{u_1, u_2\}$, $U_2 = \{w_1, w_2, w_3\}$ (see Example 2.4). We color the edges with a_1 and split every edge into two half-edges: those that are incident to the vertices in U_1 are labeled with a_1, those that are incident to the vertices in U_2 are labeled with $\hat{a}_1$. The resulting pot realizing $K_{2,3}$ is thus the set $\mathcal{T} = \{\{a_1^3\}, \{\hat{a}_1^2\}\}$. (b) The complete bipartite graph $K_{3,3}$ represented as the 3-partite graph $G = (U_1, U_2, U_3)$ with $U_1 = \{w\}$, $U_2 = \{u_1, u_2, u_3\}$ and $U_3 = \{u_2, u_3\}$ (see Example 2.5). The pot $\mathcal{T} = \{\{a_1^3\}, \{\hat{a}_1 a_2^2\}, \{\hat{a}_2^3\}\}$ realizing $K_{3,3}$ is obtained by the edge coloring that assigns the color a_i to the edges of the subgraph $B(U_i, U_{i+1})$ for $i = 1, 2$; every edge of color a_i is split into two half-edges that are labeled with a_i and $\hat{a}_i$ depending on whether they are incident to the vertices in U_i or U_{i+1}, respectively

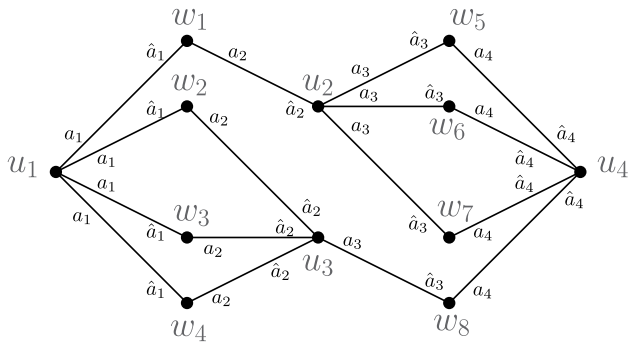

Fig. 6 The pot $\mathcal{T} = \{\{a_1^4\}, \{\hat{a}_1 a_2\}, \{\hat{a}_2 a_3^3\}, \{\hat{a}_2^3 a_3\}, \{\hat{a}_3 a_4\}, \{\hat{a}_4^4\}\}$ realizes the 5-partite graph $G = (U_1, U_2, U_3, U_4, U_5)$ in Example 2.7 with vertex partition $U_1 = \{u_1\}$, $U_3 = \{u_2, u_3\}$, $U_5 = \{u_4\}$, $U_2 = \{w_i : 1 \le i \le 4\}$ and $U_4 = \{w_i : 5 \le i \le 8\}$. $\mathcal{T}$ is obtained by assigning the color a_i to the edges of the subgraph $B(U_i, U_{i+1})$ for $1 \le i < 5$, and then by splitting every edge into two half-edges with labels a_i and $\hat{a}_i$; the half-edges that are incident to the vertices in U_i are assigned the label a_i, and the half-edges that are incident to the vertices in U_{i+1} are assigned the label $\hat{a}_i$

equation $f_{i,i+1}(X_i, X_{i+1}) = 0$, for $1 \le i \le \ell$. In this case, $f_{i,i+1}(X_i, X_{i+1}) = 0$ implies that the number of edges in $B(U_i, U_{i+1})$ that are incident to the vertices in U_i equals the number of edges in $B(U_i, U_{i+1})$ that are incident to the vertices in U_{i+1}. By the above definition of the pot $\mathcal{T}$ realizing G, this implies that the number of half-edges in $B(U_i, U_{i+1})$ that are labeled with a_i equals the number of half-edges in $B(U_i, U_{i+1})$ that are labeled with $\hat{a}_i$. Thus, if X_i represents the number of tile type $t_i \in \mathcal{T}$ used to realize the graph G, then the equations $f_{i,i+1}(X_i, X_{i+1}) = 0$ guarantee that G is a complete complex. Hence, the minimum positive value of the objective function in the integer linear programming problem in Definition 2.2 corresponds to the minimum order of a multipartite graph realized by $\mathcal{T}$. Similar observations hold for an ℓ-partite graph as in Definition 2.6.

It is thus straightforward to prove the following result.

Theorem 5.3 *Let G be an ℓ-partite graph satisfying the assumptions in Definition 2.2 (respectively, Definition 2.6). If G is a solution to the associated integer linear programming problem in Definition 2.2 (respectively, Definition 2.6), then $B_2(G) \le \ell - 1$.*

We now give a lemma that turns out to be useful in proving Corollary 5.5 and Proposition 6.1. A similar result for graphs having all vertices with even degree is proved in Almodóvar et al. (2024) using linear algebra techniques.

Lemma 5.4 *Let G be a d-regular graph of order $n \ge 2$. Then $B_2(G) > 1$ for every even value of d, with $d < n$, and for every odd value of d, with $d < n/2$.*

Proof Suppose that $B_2(G) = 1$. Then there exists a pot $\mathcal{T}$ realizing the graph G and no graph of order smaller than n, using one bond-edge type. Without loss of generality, we can write $\mathcal{T} = \{\{a^i\, \hat{a}^{d-i}\} : i \in I\}$, where I is a suitable subset of $\{0, \ldots, d\}$.

For every $i \in I$, we denote by n_i the number of vertices with tile type $\{a^i\, \hat{a}^{d-i}\}$. Since the total amount of half-edges with label a is equal to the total amount of half-edges with label $\hat{a}$, the relation $\sum_{i \in I} i\, n_i = \sum_{i \in I} (d - i) n_i$ holds. By writing the last equation in the form $\sum_{i \in I} (d - 2i) n_i = 0$, we can see that there exist two integers $h, j \in I$ such that $(d - 2h) > 0$ and $(d - 2j) < 0$.

We now construct a new graph H using the tile types $\{a^h\, \hat{a}^{d-h}\}$ and $\{a^j\, \hat{a}^{d-j}\}$ contained in $\mathcal{T}$. In the graph H, a vertex with tile type $\{a^h\, \hat{a}^{d-h}\}$ is incident to h loops and to $(d - 2h)$ half-edges with label $\hat{a}$, while a vertex with tile type $\{a^j\, \hat{a}^{d-j}\}$ is incident to $(d - j)$ loops and to $(2j - d)$ half-edges with label a. For even values of d, H has exactly $(j - d/2)$ vertices whose tile type is $\{a^h\, \hat{a}^{d-h}\}$ and $(d/2 - h)$ vertices whose tile type is $\{a^j\, \hat{a}^{d-j}\}$. For odd values of d, H has exactly $(2j - d)$ vertices whose tile type is $\{a^h\, \hat{a}^{d-h}\}$ and $(d - 2h)$ vertices whose tile type is $\{a^j\, \hat{a}^{d-j}\}$. It is easy to see that the following equalities hold: $(j - d/2)(d - 2h) + (d/2 - h)(d - 2j) = 0$ if d is even, and $(2j - d)(d - 2h) + (d - 2h)(d - 2j) = 0$ if d is odd. This means that the number of half-edges labeled with a (and not in a loop) is equal to the number of half-edges that are labeled with $\hat{a}$ (and not in a loop), that is, we can match the half-edges so as the graph H is a complete complex.

For even values of d, $d < n$, the graph H has order $(j - d/2) + (d/2 - h) = j - h \leq d < n$, since $j \leq d < n$ and $h \geq 0$. The same holds for odd values of d, $d < n/2$, since the graph H has order $(2j - d) + (d - 2h) = 2(j - h) < n$.

We find a contradiction in both cases, since $\mathcal{T}$ realizes no graph of order smaller than n. It is thus proved that $B_2(G) > 1$ for every even value of d, with $d < n$, and for every odd value of d, with $d < n/2$. □

Notice that for odd values of d, $d \geq n/2$, there exist d-regular graphs G with $B_2(G) = 1$. It is the case of the complete graph of order n with n even: in Bonvicini and Ferrari (2020) and Ellis-Monaghan et al. (2014), it is proved by different methods that $B_2(K_n) = 1$ for even values of n.

As a consequence of Theorem 5.3, the following result holds for complete bipartite graphs and special cases of bipartite graphs.

Corollary 5.5 *For the complete bipartite graph $K_{m,n}$ we have $B_2(K_{m,n}) = 1$ if $\gcd(m, n) = 1$, and $B_2(K_{m,n}) = 2$ if $\gcd(m, n) > 1$.*

For a bipartite graph $G' = (U, W)$ where $|U| = |W| = n$, $n \geq 3$, and all vertices have degree $n - 1$, we have $B_2(G') = 2$ if n is even, and $B_2(G') \leq 3$ if n is odd.

Proof It follows from Example 2.4 and Theorem 5.3 that $B_2(K_{m,n}) = 1$ if $\gcd(m, n) = 1$. By Example 2.5 and Theorem 5.3, we have that $B_2(K_{m,n}) \leq 2$ if $\gcd(m, n) > 1$. By repeating the same argument as Lemma 5.4, we can prove that $B_2(K_{m,m}) > 1$. It is thus proved that $B_2(K_{m,n}) = 2$ if $\gcd(m, n) > 1$.

Let us consider the bipartite graph $G' = (U, W)$, with $|U| = |W| = n \geq 3$ and all vertices of degree $n - 1$. By Example 3.4 and Theorem 5.3, we have that $B_2(G') \leq 3$ for odd values of n, and $B_2(G') \leq 2$ for even values of n. In the latter case, we apply Lemma 5.4 and find $B_2(G') = 2$ for every even value of n. □

We observe that the first part of Corollary 5.5 about the value of $B_2(K_{m,n})$ revisits a result already known in the literature, which is proved in Bonvicini and Ferrari (2020) and Ellis-Monaghan et al. (2014) by alternative approaches. In fact, the proof of Lemma 5.4 is a generalization of the result in Bonvicini and Ferrari (2020) about the value of the parameter $B_2(G)$ when G is either the complete graph K_n or the complete bipartite graph $K_{m,n}$.

In Bonvicini and Ferrari (2020), we proved that the relation $B_2(G) \leq B_2(H) + 1$ holds for a graph G containing a spanning subgraph H. By combining the above relation together with Theorem 5.3 and Corollary 5.5, the proof of the following result is straightforward. The result also shows that the upper bound obtained from the spanning subgraph H becomes significantly small when H is a complete bipartite graph, or a graph satisfying the assumptions in Propositions 3.1, 3.2 or 3.3.

Proposition 5.6 *Let G be a connected graph containing a spanning ℓ-partite graph H that satisfies the assumptions in Theorem 5.3. Then $B_2(G) \leq \ell$.*

In particular, it holds $B_2(G) \leq 2$ if H is the complete bipartite graph $K_{m,n}$ with $\gcd(m, n) = 1$, $B_2(G) \leq 3$ if H satisfies the assumptions in Proposition 3.2 or H is the complete bipartite graph $K_{m,n}$ with $\gcd(m, n) > 1$, $B_2(G) \leq 4$ if H satisfies the assumptions in Proposition 3.3, and $B_2(G) \leq 5$ if H satisfies the assumptions in Proposition 3.1.

We conclude this section by noting that our approach can be applied to the graphs studied in Almodóvar et al. (2024), that is, complete tripartite graphs and cocktail party graphs. However, we believe it is more interesting to show how our strategy can be applied to graphs G that are not necessarily multipartite in order to obtain bounds, or even the exact value, for $B_2(G)$.

6 Case study: the Platonic graphs

In this section, we apply our method to determine the exact value of $B_2(G)$ when the degrees of the vertices are close to half the order of the graph G. Moreover, we show that our method can provide a good upper bound for the parameter $B_2(G)$ when G is a regular graph whose degree is far from half of its order. Indeed, we apply the propositions in Sect. 3 to the graphs of the Platonic solids and find the exact value of $B_2(G)$ when the degree of the graph is at least $(|V(G)|/2) - 1$; such a condition is satisfied for all Platonic graphs G with the exception of the dodecahedral graph, which is 3-regular of order 20. For the latter graph G, we find the upper bound $B_2(G) \leq 5$, which is obtained by leveraging a particular structural property of the dodecahedral graph and is close to the bound that is known in the literature: $B_2(G) \leq 4$ (Almodovar et al. 2021) (at the time of writing, it was found that $B_2(G) = 3$ inAshworth et al. (2024)We point out that Platonic graphs are notable for DNA nanostructures as can be noted from reviews like Orponen (2018), Ke (2014), Yan et al. (2020), Ouyang et al. (2024).

Proposition 6.1 *The following relations hold:*

1. $B_2(K_4) = 1$;
2. $B_2(G) = 2$ *if G is either the cube Q_3, the octahedral or the icosahedral graph;*
3. $B_2(G) \leq 5$ *for the dodecahedral graph G.*

Proof 1. The complete graph K_4 contains the complete bipartite graph $K_{1,3}$ as a spanning subgraph. By Proposition 5.6, we have that $B_2(K_4) \leq 2$. From the pot $\mathcal{T} = \{\{a_1^3\}, \{\hat{a}_1\}\}$, constructed as described in Sect. 5, we can define a new pot $\mathcal{T}' = \{\{a_1^3\}, \{a_1\hat{a}_1^2\}\}$ that realizes the graph K_4 and no graph of order smaller than 4. It thus follows that $B_2(K_4) = 1$.

2. (a) The cube Q_3 is a 3-regular bipartite graph of order 8 and it is isomorphic to a 3-partite graph $H = (U_1, U_2, U_3)$ satisfying the assumptions in Proposition 3.2. More specifically, we select the vertices u, w as shown in Fig. 7(a) and set $U_1 = \{u, w\}$; then, we define U_2, U_3 as the neighborhoods $N(u)$, $N(w)$, of u, w, respectively. We thus have $d_1 = d_2 = d_3 = 3$, $d^1_{1,2} = d^2_{1,3} = 3$, and $d_{2,1} = d_{3,1} = 1$; whence, $\gcd(d, 2d_{2,1}) = \gcd(3, 2) = 1$ and the inequality $2(d + d_{2,1}) \geq |V(H)|$ holds true. It follows from Theorem 5.3 that $B_2(Q_3) \leq 2$; in this case, the pot realizing Q_3 is $\mathcal{T} = \{\{a_1^3\}, \{\hat{a}_1 a_2^2\}, \{\hat{a}_1\hat{a}_2^2\}\}$, which in constructed as described in Sect. 5. Since the cube Q_3 is 3-regular, we can apply Lemma 5.4 and find that $B_2(Q_3) > 1$. Therefore, $B_2(Q_3) = 2$.

(b) The octahedral graph G contains a spanning 3-partite subgraph $H = (U_1, U_2, U_3)$ satisfying the assumptions in Definition 2.2. The graph H is defined as follows: select the vertices u_i, with $1 \leq i \leq 6$, as shown in Fig. 7(b), and set $U_1 = \{u_1\}$, $U_2 = \{u_i : i = 2, 3\}$, $U_3 = \{u_i : i = 4, 5, 6\}$. The edges of H are the bolded edges in Fig. 7(b). Notice that $d_1 = d_3 = 2$, $d_2 = 4$, $d_{1,2} = d_{3,2} = 2$, $d_{2,1} = 1$, $d_{2,3} = 3$. The integer linear programming problem $ILP(H)$ associated with H is thus the following:

$$ILP(H) \begin{cases} \min(X_1 + X_2 + X_3) & \\ 2\,X_1 - X_2 & = 0 \\ 3\,X_2 - 2\,X_3 & = 0 \\ X_1 + X_2 + X_3 & > 0 \\ X_i \geq 0, & 1 \leq i \leq 3 \\ X_i \in \mathbb{Z}, & 1 \leq i \leq 3 \end{cases}$$

From the system of linear diophantine equations associated with H, we find that $X_2 = 2\,X_1$ and $X_3 = 3\,X_1$; whence, $\sum_{i=1}^{3} X_i = 6\,X_1 \geq 6$, that is, H is a solution to the associated $ILP(H)$. It follows from Proposition 5.6 that $B_2(G) \leq 3$. From the pot $\mathcal{T} = \{\{a_1^2\}, \{\hat{a}_1 a_2^3\}, \{\hat{a}_2^2\}\}$, constructed as described in Sect. 5, we can define a new pot $\mathcal{T}' = \{\{a_1^3\hat{a}_1\}, \{\hat{a}_1 a_2^3\}, \{a_1\hat{a}_1\hat{a}_2^2\}\}$ that realizes the graph G and no graph of order smaller than 6 (as we will prove in the last paragraph). It thus follows that $B_2(G) \leq 2$. Since the octahedral graph is 4-regular, we can apply Lemma 5.4 and find that $B_2(G) > 1$. Therefore, $B_2(G) = 2$. We prove that the pot $\mathcal{T}'$ realizes no graph of order smaller than 6. Similar arguments can also be repeated for the pots $\mathcal{T}'$ that are defined in the other cases that will follow. First, note that if the pot $\mathcal{T}'$ realizes a complete complex F, then there exists at least one vertex v of F whose tile type is t, for every $t \in \mathcal{T}'$. Because the tile types $t_1 = \{a_1^3\hat{a}_1\}, t_3 = \{a_1\hat{a}_1\hat{a}_2^2\}$ contain the complementary labels a_1, $\hat{a}_1$, we can obtain a new graph F' from F that has the same order as F and loops in each vertex with tile type t_1 or t_3. In fact, let v be a vertex having tile type t_i, with $i = 1, 3$, and no loops incident to v. Then there exist two edges incident to v, say $e_1 = \{v, v_1\}$ and $e_2 = \{v, v_2\}$, such that the half-edges (v, e_1) and (v, e_2) are labeled with a_1 and $\hat{a}_1$, respectively. We can delete the edges e_1, e_2, create a loop in v and add the edge $\{v_1, v_2\}$. We find a graph F' having a loop in each vertex v with tile type t_1 or t_3. Suppose $\mathcal{T}'$ realizes a complete complex F of order

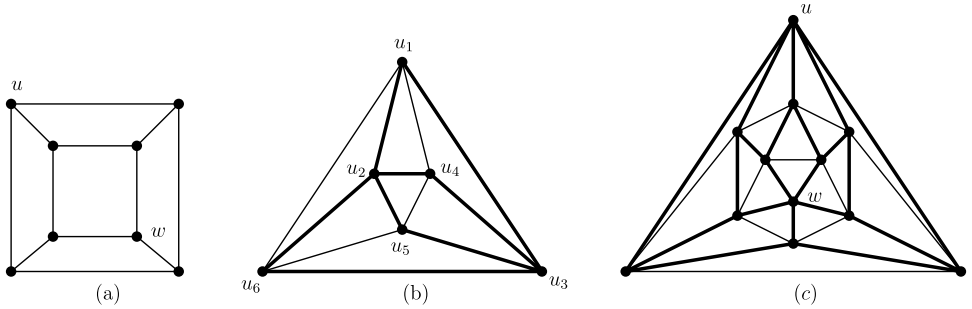

Fig. 7 The graphs of the platonic solids: (a) the cube Q_3; (b) the octahedral graph; (c) the icosahedral graph. In each graph, we select vertices as described in the proof of Proposition 6.1 and obtain an ℓ-partite spanning subgraph H given by Q_3 itself in case (a), and by the bolded edges in cases (b)-(c). In each case, the graph H satisfies the assumptions in Proposition 3.2 or Definition 2.2, and is used to determine the exact value of the parameter $B_2(G)$

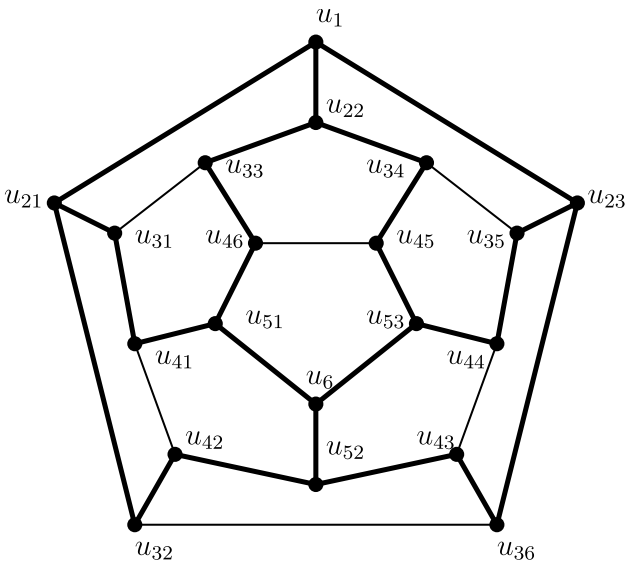

Fig.8 The dodecahedral graph and its spanning 5-partite subgraph H given by the bolded edges. The partition of the vertex set of H is defined as follows: $U_1 = \{u_1, u_6\}$, $U_2 = \{u_{2i} : 1 \le i \le 3\}$, $U_3 = \{u_{3i} : 1 \le i \le 6\}$, $U_4 = \{u_{4i} : 1 \le i \le 6\}$, $U_5 = \{u_{5i} : 1 \le i \le 3\}$. The graph H satisfies the assumptions in Definition 2.6 and is a solution to the associated *ILP*(H)

smaller than 6. Then the graph F' obtained as above has order smaller than 6. We remove the loops in F' and obtain a new graph F'' having the same order as F' and F. It is easy to see that the graph F'' is realized by the pot $\mathcal{T}$. It follows that $|V(F'')| \ge 6$, a contradiction. We have thus proved that $\mathcal{T}'$ realizes no graph of order smaller than 6.

(c) The icosahedral graph G contains a spanning 3-partite graph $H = (U_1, U_2, U_3)$ defined as follows: we select the vertices u, w as shown in Fig. 7(c) and set $U_1 = \{u, w\}$; then, we define U_2, U_3 as the neighborhoods $N(u)$, $N(w)$, of u, w, respectively. The edges of H are the bolded edges in Fig. 7(c). The graph H satisfies the assumptions in Proposition 3.2 since we have $d_1 = 5$, $d_2 = d_3 = 3$, $d^1_{1,2} = d^2_{1,3} = 5$, and $d_{2,1} = d_{3,1} = 1$; whence, $\gcd(d, d_{2,1}) = 1$ and the inequality $2(d + d_{2,1}) \ge |V(H)|$ holds true. By Proposition 5.6, we have that $B_2(G) \le 3$; in this case, the pot realizing H is $\mathcal{T} = \{\{a_1^5\}, \{\hat{a}_1 a_2^2\}, \{\hat{a}_1 \hat{a}_2^2\}\}$, which is constructed as described in Section 5. By coloring the remaining edges in G by color a_1, we find the pot $\mathcal{T}' = \{\{a_1^5\}, \{a_1 \hat{a}_1^2 a_2^2\}, \{a_1 \hat{a}_1^2 \hat{a}_2^2\}\}$ that realizes the graph G and no graph of order smaller than 12. It thus follows that $B_2(G) \le 2$. Since the icosahedral graph is 5-regular, we can apply Lemma 5.4 and find that $B_2(G) > 1$. Therefore, $B_2(G) = 2$.

3. For the dodecahedral graph G, we find the spanning 5-partite subgraph $H = (U_1, U_2, U_3, U_4, U_5)$ given by the bolded edges in Fig. 8 and vertex set partitioned as follows: $U_1 = \{u_1, u_6\}$, $U_2 = \{u_{2i} : 1 \le i \le 3\}$, $U_3 = \{u_{3i} : 1 \le i \le 6\}$, $U_4 = \{u_{4i} : 1 \le i \le 6\}$, $U_5 = \{u_{5i} : 1 \le i \le 3\}$. It is easy to see that H satisfies the assumptions in Definition 2.6 with $d_1 = d_2 = d_5 = 3$, $d_3 = d_4 = 2$, $d_{2,1} = d_{3,2} = d_{3,4} = d_{4,3} = d_{4,5} = d_{5,1} = 1$, $d_{2,3} = d_{5,4} = 2$, and $t = 1$, $d^1_{1,2} = d^2_{1,5} = 3$. The integer linear programming problem *ILP*(H) associated with H is thus the following:

$$ILP(H)\begin{cases} \min(X_1 + X_2 + X_3 + X_4 + X_5 + Y_1) & \\ 3X_1 - X_2 & = 0 \\ 2X_2 - X_3 & = 0 \\ X_3 - X_4 & = 0 \\ X_4 - 2X_5 & = 0 \\ X_5 - 3Y_1 & = 0 \\ X_1 + X_2 + X_3 + X_4 + X_5 + Y_1 & > 0 \\ Y_1 \ge 0, X_i \ge 0, & 1 \le i \le 5 \\ Y_1, X_i \in \mathbb{Z}, & 1 \le i \le 5 \end{cases}$$

From the system of linear diophantine equations associated with H, we find that $Y_1 = X_1$, $X_2 = X_5 = 3X_1$, and $X_3 = X_4 = 6X_1$; whence, $Y_1 + \sum_{i=1}^{5} X_i = 20X_1 \ge 20$, that is, H is a solution to the associated *ILP*(H). It follows from Proposition 5.6 that $B_2(G) \le 5$. □

Remark 6.2 Regarding Proposition 6.1, we note that the pot realizing the complete graph K_4 coincides with the biminimal pot provided in Ellis-Monaghan et al. (2014), Bonvicini and Ferrari (2020); the notion of biminimal pot was introduced in Ferrari et al. (2023). Nevertheless, our approach may yield pots that are not isomorphic to the ones already known in the literature. As an example, the (biminimal) pot realizing the cube Q_3 in Proposition 6.1 is not isomorphic to the one in Almodóvar et al. (2024).

7 Conclusions

In this paper, we have introduced a method that is a particular case of a more general idea consisting of studying the parameter $B_2(G)$ using a subgraph H of G whose structure can be leveraged to find a pot, possibly optimal, that realizes G in Scenario 2. In the multipartite approach presented here, such a subgraph is a spanning ℓ-partite subgraph of G. In particular, we have proved in Theorem 5.3 that $B_2(G) \le \ell - 1$ if G is an ℓ-partite graph satisfying the conditions in Definition 2.2 or Definition 2.6 on the degrees of the vertices, while in Proposition 5.6 we have proved that $B_2(G) \le \ell$ for every graph G containing a spanning ℓ-partite subgraph satisfying Definition 2.2 or Definition 2.6. The

numerical conditions in Section 3 show that the parameter ℓ can be very small, namely $\ell \leq 5$, and it corresponds to, or is very close to, the exact value of $B_2(G)$ for some graphs G. This happens in the case where G is the complete bipartite graph $K_{m,n}$ (see Corollary 5.5) or one of the Platonic graphs in Section 7. We note that the results about $B_2(G)$, when G is one of the graphs in the aforementioned families, are already known in the literature (Almodovar et al. 2021; Ellis-Monaghan et al. 2014; Almodóvar et al. 2024). However, our results emphasize the existence of a particular spanning ℓ-partite subgraph with $\ell \leq 5$ that can be leveraged to realize G in Scenario 2.

The reader may wonder how we have chosen the sets U_i in the analyzed graphs (complete bipartite graphs and Platonic graphs) or, more generally, what strategy to adopt to select the sets U_i that define a spanning ℓ-partite subgraph of a given graph G. One possible strategy, which is the one we have adopted, is to analyze the dominating sets of the graph G and choose some (or all) of the sets U_i to be subsets of a dominating set of G. Any dominating set D defines a spanning bipartite graph $B(U_1, U_2)$ of G where $U_1 = D$ and $U_2 = V(G) \setminus D$. However, the partition (U_1, U_2) defined in this way may not always give the best upper bound on the parameter $B_2(G)$. In some cases, further splitting of $U_1 = D$ into multiple subsets is more efficient.

In the future, we intend to explore different research directions. On the one hand, it would be interesting to investigate families of graphs G having a spanning ℓ-partite subgraph satisfying the conditions in Definition 2.2 or Definition 2.6. On the other hand, it would be interesting to consider other subgraphs of G whose structure can be exploited to find upper bounds for the parameter $B_2(G)$.

Acknowledgements MMF acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference numbers DGECR-2023-00131, RGPIN-2023-04722] and the University of Manitoba (research start-up funds). Simona Bonvicini is member of GNSAGA of Istituto Nazionale di Alta Matematica (INdAM).

Author contributions All authors contributed equally to this study.

Funding Open access funding provided by Università degli Studi di Modena e Reggio Emilia within the CRUI-CARE Agreement.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Almodóvar L, Ellis-Monaghan J, Harsy A, Johnson C, Sorrells J (2024) Computational complexity and pragmatic solutions for flexible tile based DNA self-assembly. Nat Comput. https://doi.org/10.1007/s11047-024-09998-x

Almodóvar L, Lee JH, Neal M, Todt H, Williams J (2024) DNA Self-assembly: complete tripartite graphs and cocktail party graphs. In: F. Hoffman, S. Holliday, Z. Rosen, F. Shahrokhi, J. Wierman (eds), Combinatorics, Graph Theory and Computing, Springer Proceedings in Mathematics & Statistics **448**

Almodóvar L, Martin S, Mauro S, Tod H (2019) Minimal tile and bond-edge types for self-assembling DNA graphs of triangular lattice graphs. Congr Numer 232:241–264

Almodovar L, Ellis-Monaghan J, Harsy A, Johnson C, Sorrells J (2021) Optimal tile-based DNA self-assembly designs for lattice graphs and Platonic solids, California State University, San Bernardino: Mathematics Faculty Publications 1. https://scholarworks.lib.csusb.edu/mathematics-publications/1

Ashworth J, Grossmann L, Navarro F, Almodovar L, Harsy A, Johnson C, Sorrells J (2024) Algorithmic pot generation: algorithms for the flexible-tile model of DNA self-assembly. arXiv preprint https://www.arxiv.org/abs/2408.00192

Bondy JA, Murty USR (2008) Graph Theory. Springer-Verlag, London

Bonvicini S, Ferrari MM (2020) On the minimum number of bond-edge types and tile types: an approach by edge-colorings of graphs. Discret Appl Math 277:1–13

Ellingham MN, Ellis-Monaghan JA (2019) Edge-outer graph embedding and the complexity of the DNA reporter strand problem. Theoret Comput Sci 785:117–127

Ellis-Monaghan J, Jonoska N (2023) From Molecules to Mathematics. In: Jonoska N, Winfree E (eds) Visions of DNA Nanotechnology at 40 for the Next 40. Natural Computing Series. Springer, Singapore

Ellis-Monaghan J, Jonoska N, Pangborn G (2019) Tile-Based DNA Nanostructures: Mathematical Design and Problem Encoding. Algebraic and Combinatorial Computational Biology Academic Press, 35-60

Ellis-Monaghan J, Pangborn G, Beaudin L, Miller D, Bruno N, Hashimoto A (2014) Minimal tile and bond-edge types for self-assembling DNA graphs. In: Jonoska N, Saito M (eds) Discrete and Topological Models in Molecular Biology. Springer, Berlin/Heidelberg, pp 241–270

Ferrari MM, Cook A, Houlihan A, Rouleau R, Seeman NC, Pangborn G, Ellis-Monaghan J (2018) Design formalism for DNA self-assembly of polyhedral skeletons using rigid tiles. J Math Chem 56(5):1365–1392

Ferrari MM, Pasotti A, Traetta T (2023) On non-isomorphic biminimal pots realizing the cube. Bull Inst Combin Appl 98:122–139

Griffin C, Sorrells J (2023) Tile-based modeling of DNA self-assembly for two graph families with appended paths. Involve a Journal of Mathematics 16(1):69–106

Hakimi SL, Kariv O (1986) A generalization of edge-coloring in graphs. J Graph Theory 10:139–154

Redmon E, Mena M, Vesta M, Renzyl Cortes A, Gernes L, Merheb S, Soto N, Stimpert C, Harsy A (2023) Optimal tilings of bipartite graphs using self-assembling DNA. The PUMP Journal of Undergraduate Research 6:124–150

Jonoska N, McColm GL, Staninska A (2011) On stoichiometry for the assembly of flexible tile DNA complexes. Nat Comput 10(3):1121–1141

Jonoska N, McColm GL (2009) Complexity classes for self-assembling flexible tiles. Theor Comput Sci 410(4):332–346

Jonoska N, McColm GL (2006) Flexible versus rigid tile assembly. In: Calude CS, Dinneen MJ, Păun G, Rozenberg G, Stepney S (eds) Unconventional Computation, Vol. 4135. Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp 139–151

Ke Y (2014) Designer three-dimensional DNA architectures. Curr Opin Struct Biol 27:122–128

Mattamira C (2020) DNA self-assembly design for gear graphs. Rose-Hulman Undergrad Math J 21(1):Article 11

Morse A, Adkisson W, Greene J, Perry D, Smith B, Ellis-Monaghan J, Pangborn G (2020) DNA origami and unknotted A-trails in torus graphs. J Knot Theory Ramifications 29(07):2050041

Orponen P (2018) Design methods for 3D wireframe DNA nanostructures. Nat Comput 17(1):147–160

Ouyang Y, Zhang P, Willner I (2024) DNA tetrahedra as functional nanostructures: from basic principles to applications. Angew Chem Int Ed 63:e202411118

Sa-Ardyen P, Jonoska N, Seeman NC (2004) Self-assembly of irregular graphs whose edges are DNA helix axes. J Am Chem Soc 126(21):6648–6657

Seeman NC (1982) Nucleic acid junctions and lattices. J Theor Biol 99(2):237–247

Wu G, Jonoska N, Seeman NC (2009) Construction of a DNA nano-object directly demonstrates computation. Biosystems 98(2):80–84

Yan X, Huang S, Wang Y, Tang Y, Tian Y (2020) Bottom-up self-assembly based on DNA nanotechnology. Nanomaterials 10(10):2047