A Novel System for Inferring of Chemical Compounds with Prescribed Topological Substructures Based on Integer Programming

Jianshen Zhu[1], Naveed Ahmed Azam[1], Fan Zhang[1], Aleksandar Shurbevski[1], Kazuya Haraguchi[1], Liang Zhao[2], Hiroshi Nagamochi[1], Tatsuya Akutsu[3]

1. Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan

2. Graduate School of Advanced Integrated Studies in Human Survivability (Shishu-Kan), Kyoto University, Kyoto 606-8306

3. Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji 611-0011, Japan

Abstract

Recently a novel framework has been proposed for design of chemical graphs using both artificial neural networks (ANNs) and mixed integer linear programming (MILP). This method consists of a prediction phase and an inverse prediction phase. In the first phase, an ANN is trained using existing chemical compound data. In the second phase, given a target chemical property, a feature vector is inferred by solving an MILP formulated from the trained ANN and then a set of chemical structures is enumerated by a graph enumeration algorithm.

Although exact solutions are guaranteed by this framework, types of chemical graphs have been restricted to such classes as trees, monocyclic graphs and graphs with a specified polymer topology with cycle index up to 2.

To overcome this limitation, we propose a new flexible modeling method to the framework so that we can specify a topological substructure of graphs and a partial assignment of chemical elements and bond-multiplicity to a target graph. The results of computational experiments suggest that the proposed system can infer chemical graphs with around up to 50 non-hydrogen atoms.

Keywords: QSAR/QSPR, Molecular design, Chemical graph, Artificial neural network, Mixed integer linear programming, Enumeration of graphs

1 Introduction

Graphs are a fundamental data structure in information science. Recently, design of novel graph structures has become a hot topic in artificial neural network (ANN) studies. In particular, extensive studies have been done on designing chemical graphs having desired chemical properties because of its potential application to drug design. For example, variational autoencoders [10], recurrent neural networks [21, 27], grammar variational autoencoders [14], generative adversarial networks [8], and invertible flow models [16, 22] have been applied.

On the other hand, computer-aided design of chemical graphs has a long history in the field of chemo-informatics, under the name of inverse quantitative structure activity/property relationships

(inverse QSAR/QSPR). In this framework, chemical compounds are usually represented as vectors of real or integer numbers, which are often called *descriptors* and correspond to *feature vectors* in machine learning. Using these chemical descriptors, various heuristic and statistical methods have been developed for finding optimal or near optimal chemical graphs [11, 17, 20]. In many of such methods, inference or enumeration of graph structures from a given set of descriptors is a crucial subtask, and thus various methods have been developed [9, 13, 15, 19]. However, enumeration in itself is a challenging task, since the number of molecules (i.e., chemical graphs) with up to 30 atoms (vertices) `C`, `N`, `O`, and `S`, may exceed 10^{60} [6]. Furthermore, even inference is a challenging task since it is NP-hard except for some simple cases [1, 18]. Indeed, most existing methods including ANN-based ones do not guarantee optimal or exact solutions.

In order to guarantee the optimality mathematically, a novel approach has been proposed [2] for ANNs, using mixed integer linear programming (MILP). However, this method outputs feature vectors only, not chemical structures. To overcome this issue, a new framework has been proposed [4, 7, 28] by combining two previous approaches; efficient enumeration of tree-like graphs [9], and MILP-based formulation of the inverse problem on ANNs [2]. This combined framework for inverse QSAR/QSPR mainly consists of two phases. The first phase solves (I) Prediction Problem, where a feature vector $f(G)$ of a chemical graph G is introduced and a prediction function $\psi_{\mathcal{N}}$ on a chemical property π is constructed with an ANN $\mathcal{N}$ using a data set of chemical compounds G and their values $a(G)$ of π. The second phase solves (II) Inverse Problem, where (II-a) given a target value y^* of the chemical property π, a feature vector x^* is inferred from the trained ANN $\mathcal{N}$ so that $\psi_{\mathcal{N}}(x^*)$ is close to y^* and (II-b) then a set of chemical structures G^* such that $f(G^*) = x^*$ is enumerated by a graph search algorithm. In (II-a) of the above-mentioned previous methods [4, 7, 28], an MILP is formulated for acyclic chemical compounds. Afterwards, Ito et al. [12] and Zhu et al. [29] designed a method of inferring chemical graphs with rank (or cycle index) 1 and 2, respectively by formulating a new MILP and using an efficient algorithm for enumerating chemical graphs with rank 1 [24] and rank 2 [25, 26]. The computational results conducted on instances with n non-hydrogen atoms show that a feature vector x^* can be inferred for up to around $n = 40$ whereas graphs G^* can be enumerated for up to around $n = 15$. Recently Azam et al. [5] introduced a new characterization of acyclic graph structure, called "branch-height" to define a class of acyclic graphs with a restricted structure that still covers the most of the acyclic chemical compounds in the database. They also employed the dynamic programming method to design a new algorithm for generating chemical acyclic graphs which now works for instances with size $n(G^*) = 50$.

The framework has been applied so far to a case of chemical compounds with a rather abstract topological structure such as acyclic or monocyclic graphs and graphs with a specified polymer topology with rank up to 2. When there is a more specific requirement on some part of the graph structure and the assignment of chemical elements in a chemical graph to be inferred, none of the above-mentioned methods can be used directly. The main reason is that generating chemical graphs from a given feature vector is a considerably hard problem: an efficient algorithm needed to be newly designed for each of different classes of graphs. In this paper, we discover a new mechanism of generating chemical graphs that can avoid the necessity that we design a new algorithm whenever a graph class changes. Based on this, we propose a new method based on

the framework so that a target chemical graph to be inferred can be specified in a more flexible way. With our specification, we can include a prescribed substructure of graphs such as a benzene ring into a target chemical graph while imposing constraints on a global topological structure of a target graph at the same time.

The paper is organized as follows. Section 2 introduces some notions on graphs, a modeling of chemical compounds and a choice of descriptors. Section 3 reviews the framework for inferring chemical compounds based on ANNs and MILPs. Section 4 introduces a method of specifying topological substructures of target chemical graphs to be inferred. Section 5 presents an idea of a formulation of an MILP that can infer a chemical graph under a given specification to target chemical graphs. Section 6 describes a new idea of generating chemical graphs that are isomorphic to a given chemical graph $G^\dagger$ in the sense that all generated chemical graphs G^* have the same feature vector $f(G^*) = f(G^\dagger)$. Section 7 reports the results on some computational experiments conducted for some chemical properties. Section 8 makes some concluding remarks. The proposed method/system is available at GitHub `https://github.com/ku-dml/mol-infer`.

2 Preliminary

This section introduces some notions and terminology on graphs, a modeling of chemical compounds and our choice of descriptors.

Let $\mathbb{R}$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the sets of reals, integers and non-negative integers, respectively. For two integers a and b, let $[a, b]$ denote the set of integers i with $a \leq i \leq b$.

Graphs Given a graph G, let $V(G)$ and $E(G)$ denote the sets of vertices and edges, respectively. The *rank* of graph G is defined to be the minimum number of edges to be removed to make the graph a tree.

A *rooted tree* is defined to be a tree where a vertex is designated as the *root*. The *height* $\mathrm{ht}(T)$ of a rooted tree T is defined to be the the maximum length of a path from the root to a leaf u For positive integers a, b and c with $b \geq 2$, let $T(a, b, c)$ denote the rooted tree such that the number of children of the root is a, the number of children of each non-root internal vertex is b and the distance from the root to each leaf is c. We see that the number of vertices in $T(a, b, c)$ is $a(b^c - 1)/(b - 1) + 1$.

Core in Cyclic Graphs Let H be a connected simple graph with rank at least 1. The *core* $\mathrm{Cr}(H)$ of H is defined to be an induced subgraph $\mathrm{Cr}(H) = (V' = V'_1 \cup V'_2, E')$ such that V'_1 is the set of vertices in a cycle of H and V'_2 is the set of verices each of which is in a path between two vertices $u, v \in V'_1$. A vertex (resp., an edge) in H is called a *core-vertex* (resp., *core-edge*) if it is contained in the core $\mathrm{Cr}(H)$ and is called a *non-core-vertex* (resp., *non-core-edge*) otherwise. The *core size* $\mathrm{cs}(H)$ is defined to be the number of core-vertices in the core of H.

Let $H - E^{\mathrm{co}}$ denote the graph obtained from H by removing all core-edges. We call a connected component T with at least one edge in $H - E^{\mathrm{co}}$ an *exterior-tree*, which contain exactly one core-vertex v of H, where T is regarded as a rooted tree rooted at the core-vertex v. The *core height* $\mathrm{ch}(H)$ is defined to be the maximum height $\mathrm{ht}(T)$ of an exterior-tree T of H. Figure 1 illustrates

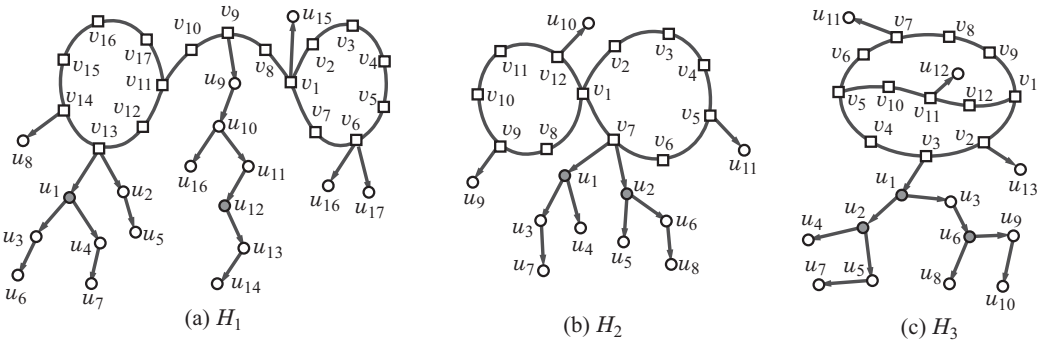


Figure 1: An illustration of rank-2 graphs H_i, $i = 1, 2, 3$.

three examples of rank-2 graphs H_i, $i = 1, 2, 3$, where $\mathrm{cs}(H_1) = 17$, $\mathrm{ch}(H_1) = 6$, $\mathrm{cs}(H_2) = 12$, $\mathrm{ch}(H_2) = 3$, $\mathrm{cs}(H_3) = 12$ and $\mathrm{ch}(H_3) = 5$.

Branch-parameter Azam et al. [5] introduced "branch-parameter," a positive integer ρ to measure the "agglomeration degree" of trees.

A *leaf ρ-branch* is defined to be a non-root vertex v in an exterior-tree T such that $\mathrm{height}(v) = \rho$. The *$\rho$-branch-leaf-number* $\mathrm{bl}_\rho(H)$ of H is defined to the number of leaf ρ-branches in H.

A non-core edge (resp., a non-core vertex) in an exterior-tree T is called an *ρ-internal edge* (resp., *ρ-internal vertex*) if it is contained in a path from the root of T to a leaf ρ-branch and is called an *ρ-external edge* (resp., *ρ-external vertex*) otherwise.

A *ρ-fringe-tree* is defined to be a maximal subtree T' of an exterior-tree T such that the edge set of T' consists of ρ-external edges.

We call a graph H *ρ-lean* if the set of ρ-internal edges in each exterior-tree T forms a single path. For $\rho = 2$, nearly 97% of cyclic chemical compounds with up to 100 non-hydrogen atoms in PubChem are 2-lean. For the graph H_1 in Figure 1(a), u_1 and u_{12} are the leaf 2-branches, and H_1 is 2-lean. For the graph H_2 in Figure 1(b), u_1 and u_2 are the leaf 2-branches, and H_2 is not 2-lean. For the graph H_3 in Figure 1(c), and H_3 is not 2-lean.

2.1 Modeling of Chemical Compounds

We represent the graph structure of a chemical compound as a graph H with labels on vertices and multiplicity on edges in a hydrogen-suppressed model. We treat a cyclic graph H as a graph possibly with undirected and directed edges by regarding each non-core-edge $uv \in E$ as a directed edge (u, v) such that u is the parent of v in some pendant-tree of H.

Let Λ be a set of labels each of which represents a chemical element such as `C` (carbon), `O` (oxygen), `N` (nitrogen) and so on. Let mass(`a`) and val(`a`) denote the mass and valence of a chemical element $\mathtt{a} \in \Lambda$, respectively. A tuple $(\mathtt{a}, \mathtt{b}, m)$ with chemical elements $\mathtt{a}, \mathtt{b}$ and a bond-multiplicity m, called an *adjacency-configuration* was used to represent a pair of atoms `a` and `b` joined by a bond-multiplicity m [7]. In this paper, we introduce "edge-configuration," a refined notion of adjacency-configuration.

We call a pair $(\mathtt{a}, i)$ of the chemical element `a` and the degree i a *chemical symbol*. Let Λ_{dg}

denote the set of all chemical symbols. We call a tuple $(\mathtt{a}i, \mathtt{b}j, m)$ with $\mathtt{a}i, \mathtt{b}j \in \Lambda_{\rm dg}$ and $m \in [1,3]$ an *edge-configuration*. We choose a branch-parameter ρ and two sets $\Lambda_{\rm dg}^{\rm co}$ and $\Lambda_{\rm dg}^{\rm nc}$ of chemical symbols and three sets $\Gamma^{\rm co}$, $\Gamma^{\rm in}$, $\Gamma^{\rm ex}$ of edge-configurations.

Let $e = uv$ be an edge in a chemical graph G such that $\mathtt{a}, \mathtt{b} \in \Lambda$ are assigned to the vertices u and v, the degrees of u and v are i and j, respectively and the bond-multiplicity between them is m. When uv is a core-edge, the edge-configuration $\tau(e)$ of edge e is defined to be $(\mathtt{a}i, \mathtt{b}j, m)$ if $\mathtt{a}i \leq \mathtt{b}j$ in a total order over $\Lambda_{\rm dg}$ (or $(\mathtt{b}j, \mathtt{a}i, m)$ otherwise). When uv is a non-core-edge which is regarded as a directed edge (u, v) where u is the parent of v in some pendant-tree, the edge-configuration $\tau(e)$ of a ρ-internal (resp., ρ-external) edge e is defined to be $(\mathtt{a}i, \mathtt{b}j, m) \in \Gamma^{\rm in}$ (resp., $(\mathtt{a}i, \mathtt{b}j, m) \in \Gamma^{\rm ex}$).

A *chemical cyclic graph* is defined to be a tuple $G = (H, \alpha, \beta)$ of a cyclic graph $H = (V, E)$ and functions $\alpha : V \to \Lambda$ and $\beta : E \to [1,3]$ such that (i) H is connected; (ii) $\sum_{uv \in E} \beta(uv) \leq {\rm val}(\alpha(u))$ for each vertex $u \in V$; and (iii) $\tau(e) \in \Gamma^{\rm co}$, $\tau(e) \in \Gamma^{\rm in}$ and $\tau(e) \in \Gamma^{\rm ex}$ for each core-edge $e \in E$, ρ-internal edge $e \in E$, $\tau(e) \in \Gamma^{\rm ex}$, and ρ-external edge $e \in E$, respectively.

In our method, we use only graph-theoretical descriptors for defining a feature vector. A *feature vector* $f(G)$ of a chemical cyclic graph $G = (H = (V, E), \alpha, \beta)$ consists of the following 16 kinds of descriptors.

- $n(G)$: the number $|V|$ of vertices; ${\rm cs}(G)$: the core size of G; ${\rm ch}(G)$: the core height of G; ${\rm bl}_\rho(G)$: the ρ-branch-leaf-number of G;

- $\overline{\rm ms}(G)$: the average mass of atoms in G; ${\rm ns}_{\mathtt{H}}(G)$: the number of hydrogen atoms suppressed in G;

- ${\rm dg}_i^{\rm co}(G)$, ${\rm dg}_i^{\rm nc}(G)$: the numbers of core-vertices and non-core-vertices of degree $i \in [1,4]$ in G;

- ${\rm bd}_m^{\rm co}(G)$, ${\rm bd}_m^{\rm in}(G)$, ${\rm bd}_m^{\rm ex}(G)$: the numbers of core-edges, ρ-internal edges and ρ-external edges with bond multiplicity $m \in [2,3]$ in G;

- ${\rm ns}_\mu^{\rm co}(G)$, $\mu \in \Lambda_{\rm dg}^{\rm co}$, ${\rm ns}_\mu^{\rm nc}(G)$, $\mu \in \Lambda_{\rm dg}^{\rm nc}$: the numbers of core-vertices and non-core-vertices v with $\alpha(v) = \mathtt{a}$ and degree i for $\mu = \mathtt{a}i$; and

- ${\rm ec}_\gamma^{\rm co}(G)$, ${\rm ec}_\gamma^{\rm in}(G)$, ${\rm ec}_\gamma^{\rm ex}(G)$: the numbers of core-edges $e \in E$ such that $\tau(e) = \gamma \in \Gamma^{\rm co}$, ρ-internal edges $e \in E$ such that $\tau(e) = \gamma \in \Gamma^{\rm in}$, and ρ-external edges $e \in E$ such that $\tau(e) = \gamma \in \Gamma^{\rm ex}$ in G.

3 A Framework for the Inverse QSAR/QSPR

We review the framework that solves the inverse QSAR/QSPR by using MILPs [7, 12, 29], which consists of two phases as illustrated in Figure 2.

Phase 1.

For a specified chemical property π such as boiling point, we denote by $a(G)$ the observed value of the property π for a chemical compound G. As the first phase, we solve (I) Prediction Problem

with the following three stages.

Stage 1: Let DB be a set of chemical graphs. For a specified chemical property π, choose a class $\mathcal{G}$ of graphs such as acyclic graphs or graphs with a given rank r. Prepare a data set $D_\pi = \{G_i \mid i = 1, 2, \ldots, m\} \subseteq \mathcal{G} \cap \mathrm{DB}$ such that the value $a(G_i)$ of each chemical graph G_i, $i = 1, 2, \ldots, m$ is available. Set reals $\underline{a}, \overline{a} \in \mathbb{R}$ so that $\underline{a} \leq a(G_i) \leq \overline{a}$, $i = 1, 2, \ldots, m$.

Stage 2: Introduce a feature function $f : \mathcal{G} \to \mathbb{R}^K$ for a positive integer K. We call $f(G)$ the *feature vector* of $G \in \mathcal{G}$, and call each entry of a vector $f(G)$ a *descriptor* of G.

Stage 3: Construct a prediction function $\psi_{\mathcal{N}}$ with an ANN $\mathcal{N}$ that, given a vector in $\mathbb{R}^K$, returns a real in the range $[\underline{a}, \overline{a}]$ so that $\psi_{\mathcal{N}}(f(G))$ takes a value nearly equal to $a(G)$ for many chemical graphs in D. Akutsu and Nagamochi [2] formulated an MILP $\mathcal{M}(x, y; \mathcal{C}_1)$ with variable vectors $x \in \mathbb{R}^K$, $y \in \mathbb{R}$, and an auxiliary variable vector $z \in \mathbb{R}^p$ for some integer p and a set $\mathcal{C}_1$ of constraints on these variables such that: $\psi_{\mathcal{N}}(x^*) = y^*$ if and only if there is a vector (x^*, y^*) feasible to $\mathcal{M}(x, y; \mathcal{C}_1)$.

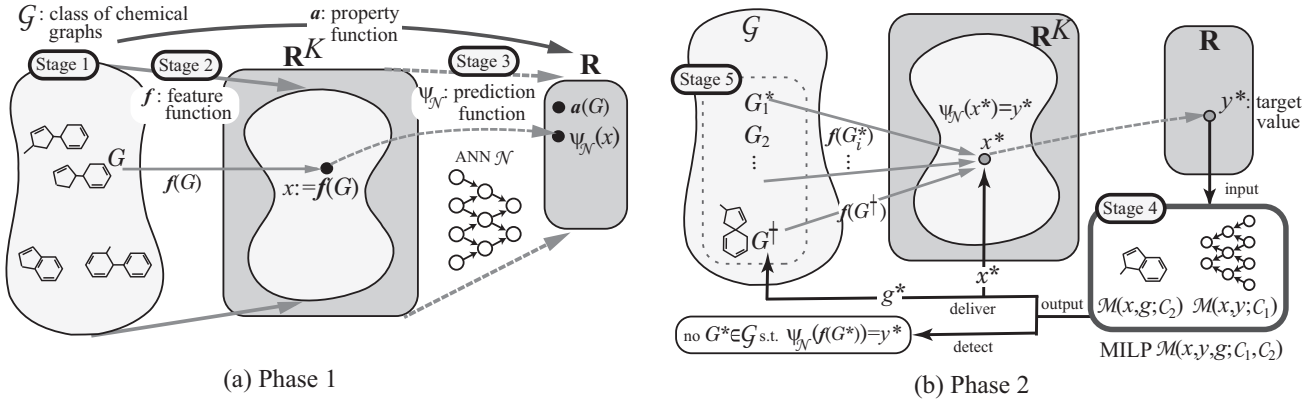


Figure 2: (a) An illustration of Phase 1; (b) An illustration of Phase 2.

Phase 2.

Given a target chemical value y^*, the second phase first finds a vector $x^* \in \mathbb{R}^K$ such that $\psi_{\mathcal{N}}(x^*) = y^*$ and then generates chemical graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$. The second phase consists of the next two stages.

Stage 4: For a real y^* with $\underline{a} \leq y^* \leq \overline{a}$, we consider the problem of finding vectors $x^* \in \mathbb{R}^K$ and $g^* \in \mathbb{R}^q$ such that $\psi_{\mathcal{N}}(x^*) = y^*$ and g^* forms a chemical graph $G^\dagger \in \mathcal{G}$ with $f(G^\dagger) = x^*$. Based on the method due to Akutsu and Nagamochi [2], Azam et al. [4], Ito et al. [12] and Zhu et al. [29] showed that this problem can be formulated as an MILP $\mathcal{M}(x, y, g; \mathcal{C}_1, \mathcal{C}_2)$ with a set $\mathcal{C}_2$ of constraints for the case of chemical acyclic graphs, chemical monocyclic graphs and chemical rank-2 graphs, respectively. The task of Stage 4 is to find a feasible solution (x^*, g^*) to the MILP such that $x^* \in \mathbb{R}^K$ and $\psi_{\mathcal{N}}(x^*) = y^*$.

Stage 5: Given a vector $x^* \in \mathbb{R}^K$ inferred in Stage 4, the task of Stage 5 is to generate all (or a specified number) of graphs $G^* \in \mathcal{G}$ such that $f(G^*) = x^*$ for the vector x^*.

4 Specifying Target Chemical Graphs

This section presents a flexible way of specifying a topological structure of the core and assignments of chemical elements and bond-multiplicity of a target chemical graph. We define a *target specification* $(G_{\rm C}, \sigma_{\rm co}, \sigma_{\rm nc}, \sigma_{\alpha\beta})$ with a multigraph $G_{\rm C}$ and sets $\sigma_{\rm co}, \sigma_{\rm nc}$ and $\sigma_{\alpha\beta}$ of lower and upper bounds on several descriptors that we describe in the following.

Seed Graphs A *seed graph* $G_{\rm C} = (V_{\rm C}, E_{\rm C})$ is defined to be a multigraph with no self-loops such that the edge set $E_{\rm C}$ consists of four sets $E_{(\geq 2)}$, $E_{(\geq 1)}$, $E_{(0/1)}$ and $E_{(=1)}$. Figure 3(a) illustrates an example of a seed graph. From a seed graph $G_{\rm C}$, the core of a cyclic graph will be constructed in the following way: Each edge $e = uv \in E_{(\geq 2)}$ will be replaced with a u, v-path P_e of length at least 2; Each edge $e = uv \in E_{(\geq 1)}$ will be replaced with a u, v-path P_e of length at least 1; Each edge $e \in E_{(0/1)}$ is either used or discarded; and Each edge $e \in E_{(=1)}$ is always used directly.

Core Specification The core of a target chemical graph is constructed from a seed graph $G_{\rm C}$ by a *core specification* $\sigma_{\rm co}$ that consists of the following:

- Lower and upper bounds $\mathrm{cs}_{\rm LB}, \mathrm{cs}_{\rm UB} \in \mathbb{Z}_+$ on the core size, where we assume $\mathrm{cs}_{\rm LB} \geq |V_{\rm C}| + \sum_{e \in E_{(\geq 2)} \cup E_{(\geq 1)}} (\ell_{\rm LB}(e) - 1)$; and
- Lower and upper bound functions $\ell_{\rm LB}, \ell_{\rm UB} : E_{(\geq 2)} \cup E_{(\geq 1)} \to \mathbb{Z}_+$; For a notational convenience, set $\ell_{\rm LB}(e) := 0$, $\ell_{\rm UB}(e) := 1$, $e \in E_{(0/1)}$ and $\ell_{\rm LB}(e) := 1$, $\ell_{\rm UB}(e) := 1$, $e \in E_{(=1)}$.

Example 1 of $\sigma_{\rm co}$. A core specification $\sigma_{\rm co}$ to $G_{\rm C}$ in Figure 3(a) is given as follows: $\mathrm{cs}_{\rm LB} = 20$, $\mathrm{cs}_{\rm UB} = 28$ and a sequence of $(\ell_{\rm LB}(a_i), \ell_{\rm LB}(a_i)), i \in [1, 6]$ is given by $[(2, 3), (2, 4), (2, 3), (3, 5), (2, 4), (1, 4)]$.

A $\sigma_{\rm co}$-*extension* of a seed graph $G_{\rm C}$ is defined to be a graph C with $|V(C)| \in [\mathrm{cs}_{\rm LB}, \mathrm{cs}_{\rm UB}]$ obtained from replacing each edge $e = uv \in E_{(\geq 2)} \cup E_{(\geq 1)}$ with a u, v-path P_e of length $\ell(P_e) \in [\ell_{\rm LB}(e), \ell_{\rm UB}(e)]$. Let C be a $\sigma_{\rm co}$-extension of $G_{\rm C}$, where each edge $e = uv \in E_{(\geq 2)} \cup E_{(\geq 1)}$ is replaced with a u, v-path P_e (where possibly P_e is equal to e). For each edge $e = uv \in E_{(\geq 2)} \cup E_{(\geq 1)}$ let $\mathcal{F}(P_e)$ denote the set of trees T_w rooted at internal vertices w of the u, v-path P_e (where $w \neq u, v$).

Figure 3(b) illustrates an $\sigma_{\rm co}$-extension C of $G_{\rm C}$ in Figure 3(a) with core specification $\sigma_{\rm co}$ in Example 1, where the edge $a_7 \in E_{(0/1)}$ is discarded.

Non-core Specification We next construct a ρ-lean cyclic graph H obtained from a $\sigma_{\rm co}$-extension C of $G_{\rm C}$, by appending a tree T_v with at most one leaf ρ-branch at each vertex $v \in V(C)$, where possibly $E(T_v) = \emptyset$. We call the vertices in C *core-vertices* of H and the newly added vertices *non-core-vertices* of H.

We specify the structure of the non-core part of such a graph H by a *non-core specification* $\sigma_{\rm nc}$ that consists of the following:

- Lower and upper bounds $n_{\rm LB}, n^* \in \mathbb{Z}_+$ on the number of vertices, where $\mathrm{cs}_{\rm LB} \leq n_{\rm LB} \leq n^*$;
- An upper bound $\mathrm{dg}^{\rm nc}_{4,\rm UB} \in \mathbb{Z}_+$ on the number of non-core-vertices of degree 4;

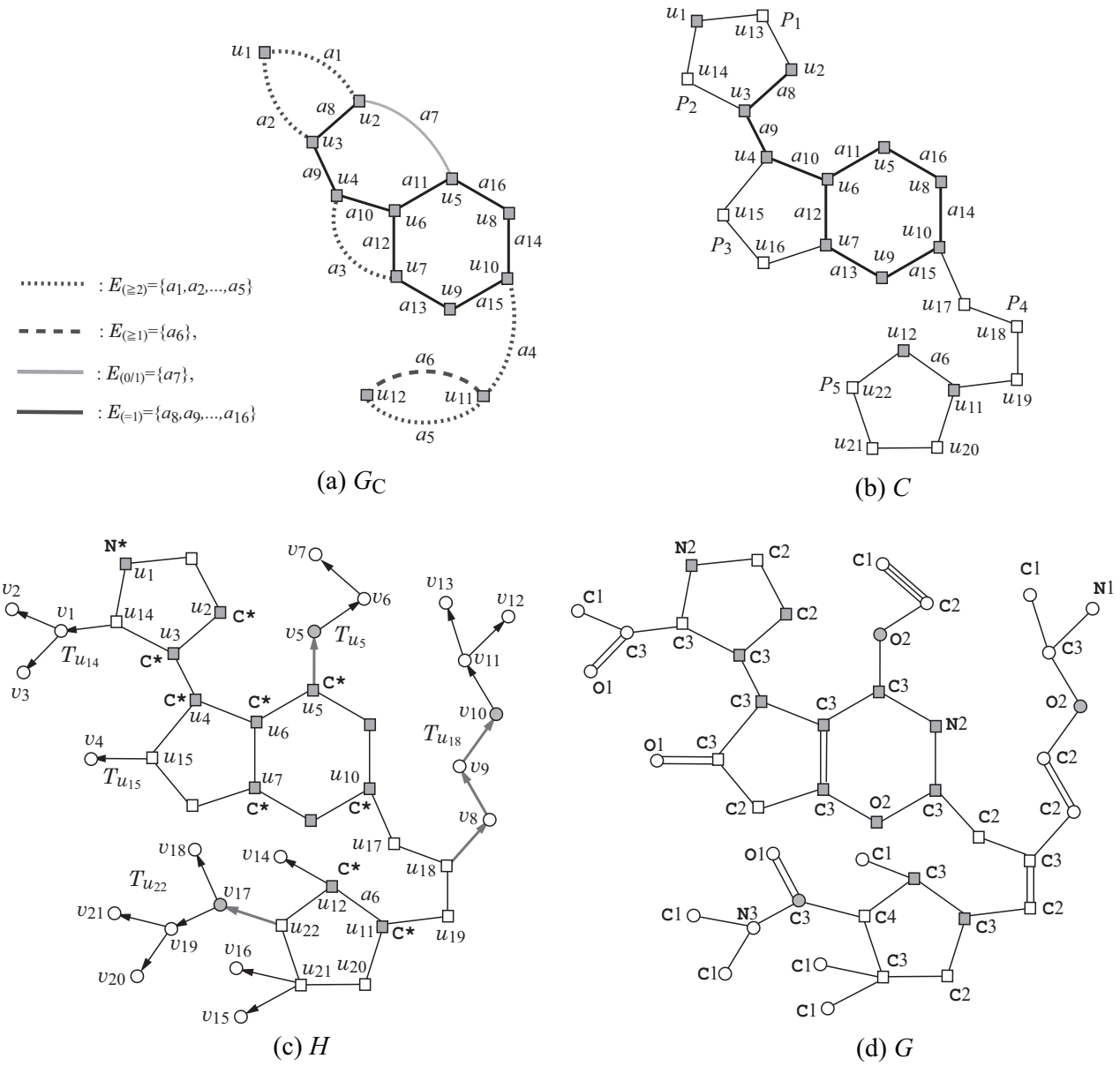


Figure 3: (a) A seed graph G_{C}; (b) A σ_{co}-extension C with $\mathrm{cs}(C) = 22$; (c) A $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$-extension H with $\mathrm{Cr}(H) = C$ with $n(H) = 43$, $\mathrm{ch}(H) = 5$ and $\mathrm{bl}_2(H) = 3$; (d) A $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$-extension G of G_{C} in Figure 3(a).

- Lower and upper functions $\mathrm{ch}_{\mathrm{LB}}, \mathrm{ch}_{\mathrm{UB}} : V_{\mathrm{C}} \to \mathbb{Z}_+$ and $\mathrm{ch}_{\mathrm{LB}}, \mathrm{ch}_{\mathrm{UB}} : E_{(\geq 2)} \cup E_{(\geq 1)} \to \mathbb{Z}_+$ on the maximum height of trees rooted at a vertex $v \in V_{\mathrm{C}}$ or at an internal vertex of a path P_e with $e \in E_{(\geq 2)} \cup E_{(\geq 1)}$;

- A branch-parameter $\rho \in \mathbb{Z}_+$; and

- Lower and upper functions $\mathrm{bl}_{\mathrm{LB}}, \mathrm{bl}_{\mathrm{UB}} : V_{\mathrm{C}} \to \{0, 1\}$ on the number of leaf ρ-branches in the tree rooted at a vertex $v \in V_{\mathrm{C}}$;
  Lower and upper functions $\mathrm{bl}_{\mathrm{LB}}, \mathrm{bl}_{\mathrm{UB}} : E_{(\geq 2)} \cup E_{(\geq 1)} \to \mathbb{Z}_+$ on the number of leaf ρ-branches

in the trees rooted at internal vertices in a path P_e constructed for an edge $e \in E_{(\geq 2)} \cup E_{(\geq 1)}$.

Example 2 of $\sigma_{\rm nc}$. A non-core specification $\sigma_{\rm nc}$ to $G_{\rm C}$ in Figure 3(a) is given as follows: $n_{\rm LB} = 30$, $n^* = 50$, $\rho = 2$, a sequence of $(\mathrm{ch}_{\rm LB}(u_i), \mathrm{ch}_{\rm LB}(u_i)), i \in [1, 12]$ is given by $[(0,1),(0,0),(0,0),(0,0),(1,3),(0,0),(0,1),(0,1),(0,0),(0,1),(0,2),(0,4)]$, a sequence of $(\mathrm{ch}_{\rm LB}(a_i), \mathrm{ch}_{\rm LB}(a_i)), i \in [1, 6]$ is given by $[(0,3),(1,3),(0,1),(4,6),(3,5),(0,2)]$, a sequence of $(\mathrm{bl}_{\rm LB}(u_i), \mathrm{bl}_{\rm LB}(u_i)), i \in [1, 12]$ is given by $[(0,1),(0,1),(0,1),(0,1),(0,1),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0),(0,0)]$ and a sequence of $(\mathrm{bl}_{\rm LB}(a_i), \mathrm{bl}_{\rm LB}(a_i)), i \in [1, 6]$ is given by $[(0,1),(0,1),(0,0),(1,2),(1,1),(0,0)]$.

We call the above ρ-lean cyclic graph H a $(\sigma_{\rm co}, \sigma_{\rm nc})$-*extension* of $G_{\rm C}$ if the following hold: $n(H) \in [n_{\rm LB}, n^*]$; $\mathrm{dg}^{\rm nc}_4(H) \leq \mathrm{dg}^{\rm nc}_{4,\rm UB}$; For each vertex $v \in V_{\rm C}$, the tree T_v attached to v satisfies $\mathrm{ht}(T_v) \in [\mathrm{ch}_{\rm LB}, \mathrm{ch}_{\rm UB}]$; For each edge $e \in E_{(\geq 2)} \cup E_{(\geq 1)}$, $\max\{\mathrm{ht}(T) \mid T \in \mathcal{F}(P_e)\} \in [\mathrm{ch}_{\rm LB}(e), \mathrm{ch}_{\rm UB}(e)]$; Each tree T_v, $v \in V(C)$ contains at most one leaf ρ-branch; and For each edge $e \in E_{(\geq 2)} \cup E_{(\geq 1)}$, $\sum\{\mathrm{bl}_\rho(T) \mid T \in \mathcal{F}(P_e)\} \in [\mathrm{bl}_{\rm LB}(e), \mathrm{bl}_{\rm UB}(e)]$.

Figure 3(c) illustrates a $(\sigma_{\rm co}, \sigma_{\rm nc})$-extension H of $G_{\rm C}$ in Figure 3(a) with the specification $(\sigma_{\rm co}, \sigma_{\rm nc})$ in Examples 1 and 2.

Chemical Specification To infer a chemical graph $G = (H, \alpha, \beta)$ from a $(\sigma_{\rm co}, \sigma_{\rm nc})$-extension H of $G_{\rm C}$, we finally a way of assigning elements in Λ and bond-multiplicities by a *chemical specification* $\sigma_{\alpha\beta}$ that consists of the following:

- Sets $\Lambda^{\rm co}, \Lambda^{\rm nc}$ of chemical elements. For a chemical graph G, let $\mathrm{na}_{\mathtt{a}}(G)$ (resp., $\mathrm{na}^{\rm co}_{\mathtt{a}}(G)$ and $\mathrm{na}^{\rm nc}_{\mathtt{a}}(G)$) denote the number of vertices (resp., core-vertices and non-core-vertices) in G assigned chemical element $\mathtt{a} \in \Lambda$ (resp., $\mathtt{a} \in \Lambda^{\rm co}$ and $\mathtt{a} \in \Lambda^{\rm nc}$);

- Sets $\Lambda^{\rm co}_{\rm dg}, \Lambda^{\rm nc}_{\rm dg}$ of chemical symbols and $\Gamma^{\rm co}$, $\Gamma^{\rm in}, \Gamma^{\rm ex}$ of edge-configurations;

- Set the following sets of adjacency-configurations:

$$\Gamma^{\rm co}_{\rm ac} := \{\mathrm{ac}(\gamma) \mid \gamma \in \Gamma^{\rm co}\}, \Gamma^{\rm in}_{\rm ac} := \{\mathrm{ac}(\gamma) \mid \gamma \in \Gamma^{\rm in}\}, \Gamma^{\rm ex}_{\rm ac} := \{\mathrm{ac}(\gamma) \mid \gamma \in \Gamma^{\rm ex}\}.$$

  Define the adjacency-configuration of a core-edge uv to be $(\mathtt{a}, \mathtt{b}, \beta(uv))$ with $\{\mathtt{a}, \mathtt{b}\} = \{\alpha(u), \alpha(v)\}$ and the adjacency-configuration of a directed non-core edge (u, v) to be $(\alpha(u), \alpha(v), \beta(uv))$. Let $\mathrm{ac}^{\rm co}_\nu(G)$ (resp., $\mathrm{ac}^{\rm in}_\nu(G)$ and $\mathrm{ac}^{\rm ex}_\nu(G)$) denote the number of core-edges (resp., directed ρ-internal edges and directed ρ-external edges) in G assigned adjacency-configuration $\nu \in \Gamma^{\rm co}_{\rm ac}$ (resp., $\nu \in \Gamma^{\rm in}_{\rm ac}$ and $\nu \in \Gamma^{\rm ex}_{\rm ac}$);

- Subsets $\Lambda^*(v)$, $v \in V_{\rm C}$ of elements that are allowed to be assigned to vertex $v \in V_{\rm C}$;

- Lower and upper bound functions $\mathrm{na}_{\rm LB}, \mathrm{na}_{\rm UB} : \Lambda \to [1, n^*]$ and $\mathrm{na}^{\rm t}_{\rm LB}, \mathrm{na}^{\rm t}_{\rm UB} : \Lambda^{\rm t} \to [1, n^*]$ (resp., $\mathrm{ns}_{\rm LB}, \mathrm{ns}_{\rm UB} : \Lambda_{\rm dg} \to [1, n^*]$ and $\mathrm{ns}^{\rm t}_{\rm LB}, \mathrm{ns}^{\rm t}_{\rm UB} : \Lambda^{\rm t}_{\rm dg} \to [1, n^*]$), $\mathrm{t} \in \{\mathrm{co}, \mathrm{nc}\}$ on the number of core-vertices and non-core-vertices, respectively, assigned chemical element $\mathtt{a}$ (resp., chemical symbol μ);

- Lower and upper bound functions $\mathrm{ac}^{\rm t}_{\rm LB}, \mathrm{ac}^{\rm t}_{\rm UB} : \Gamma^{\rm t}_{\rm ac} \to \mathbb{Z}_+$ (resp., $\mathrm{ec}^{\rm t}_{\rm LB}, \mathrm{ec}^{\rm t}_{\rm UB} : \Gamma^{\rm t} \to \mathbb{Z}_+$), $\mathrm{t} \in \{\mathrm{co}, \mathrm{in}, \mathrm{ex}\}$ on the number of core-edges, directed ρ-internal edges and directed ρ-external edges, respectively, assigned adjacency-configuration ν (resp., edge-configurations γ); and

- Lower and upper functions $\mathrm{bd}_{m,\mathrm{LB}}, \mathrm{bd}_{m,\mathrm{UB}} : E_{\mathrm{C}} \to \mathbb{Z}_+$, $m \in [2,3]$, where $\mathrm{bd}_{2,\mathrm{LB}}(e) + \mathrm{bd}_{3,\mathrm{LB}}(e) \leq \ell_{\mathrm{UB}}(e)$, $e \in E_{\mathrm{C}}$.

Example 3 of $\sigma_{\alpha\beta}$. A chemical specification $\sigma_{\alpha\beta}$ to G_{C} in Figure 3(a) is given by $\Lambda = \{\mathtt{C}, \mathtt{N}, \mathtt{O}\}$;
$\Lambda^{\mathrm{co}}_{\mathrm{dg}} = \{\mathtt{C2}, \mathtt{C3}, \mathtt{C4}, \mathtt{N2}, \mathtt{O2}\}$; $\Lambda^{\mathrm{nc}}_{\mathrm{dg}} = \Lambda^{\mathrm{co}}_{\mathrm{dg}} \cup \{\mathtt{C1}, \mathtt{N1}, \mathtt{O1}\}$; $\Gamma^{\mathrm{in}}_{\mathrm{ac}} = \{(\mathtt{C}, \mathtt{C}, 1), (\mathtt{C}, \mathtt{C}, 2), (\mathtt{C}, \mathtt{O}, 1)\}$;
$\Gamma^{\mathrm{co}}_{\mathrm{ac}} = \Gamma^{\mathrm{in}}_{\mathrm{ac}} \cup \{(\mathtt{C}, \mathtt{N}, 1)\}$; $\Gamma^{\mathrm{ex}}_{\mathrm{ac}} = \{(\mathtt{C}, \mathtt{C}, 1), (\mathtt{C}, \mathtt{C}, 3), (\mathtt{C}, \mathtt{N}, 1), (\mathtt{N}, \mathtt{C}, 1), (\mathtt{C}, \mathtt{O}, 1), (\mathtt{C}, \mathtt{O}, 2), (\mathtt{O}, \mathtt{C}, 1)\}$;
$\Gamma^{\mathrm{co}} = \{(\mathtt{C2}, \mathtt{C2}, 1), (\mathtt{C2}, \mathtt{C3}, 1), (\mathtt{C2}, \mathtt{C3}, 2), (\mathtt{C2}, \mathtt{C4}, 1), (\mathtt{C3}, \mathtt{C3}, 1), (\mathtt{C3}, \mathtt{C3}, 2), (\mathtt{C3}, \mathtt{C4}, 1), (\mathtt{C2}, \mathtt{N2}, 1), (\mathtt{C3}, \mathtt{N2}, 1), (\mathtt{C3}, \mathtt{O2}, 1)\}$;
$\Gamma^{\mathrm{in}} = \{(\mathtt{C2}, \mathtt{C2}, 2), (\mathtt{C3}, \mathtt{C2}, 1), (\mathtt{C3}, \mathtt{C3}, 1), (\mathtt{C2}, \mathtt{O2}, 1), (\mathtt{C3}, \mathtt{O2}, 1)\}$;
$\Gamma^{\mathrm{ex}} = \{(\mathtt{C3}, \mathtt{C1}, 1), (\mathtt{C2}, \mathtt{C1}, 3), (\mathtt{C3}, \mathtt{C3}, 1), (\mathtt{C4}, \mathtt{C1}, 1), (\mathtt{C3}, \mathtt{N1}, 1), (\mathtt{C3}, \mathtt{N3}, 1), (\mathtt{C3}, \mathtt{O1}, 2), (\mathtt{O2}, \mathtt{C2}, 1), (\mathtt{O2}, \mathtt{C3}, 1), (\mathtt{N3}, \mathtt{C1}, 1)\}$;
$\Lambda^*(u_1) = \{\mathtt{N}\}$, $\Lambda^*(u_8) = \{\mathtt{C}, \mathtt{N}\}$, $\Lambda^*(u_9) = \{\mathtt{C}, \mathtt{O}\}$, $\Lambda^*(u) = \{\mathtt{C}\}$, $u \in V_{\mathrm{C}} \setminus \{u_1, u_8, u_9\}$. We omit showing examples of lower and upper bounds (see the detailed preprint version [3] for examples of them).

A $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$-*extension* of G_{C} is a chemical graph $G = (H, \alpha, \beta)$ for a graph $H \in \mathcal{H}(G_{\mathrm{C}}, \sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$ such that $\sum_{uv \in E} \beta(uv) \leq \mathrm{val}(\alpha(u))$ for each vertex $u \in V(H)$; $\tau(e) \in \Gamma^{\mathrm{co}}$ for each core-edge e; $\tau(e) \in \Gamma^{\mathrm{in}}$ for each directed ρ-internal edge; $\tau(e) \in \Gamma^{\mathrm{ex}}$ for each directed ρ-external edge; $\alpha(v) \in \Lambda^*(v)$ for each vertex $v \in V_{\mathrm{C}}$; and the specified lower and upper bounds are satisfied; i.e., $\mathrm{na}_{\mathtt{a}}(G) \in [\mathrm{na}_{\mathrm{LB}}(\mathtt{a}), \mathrm{na}_{\mathrm{UB}}(\mathtt{a})]$, $\mathtt{a} \in \Lambda$, $\mathrm{na}^{\mathrm{co}}_{\mathtt{a}}(G) \in [\mathrm{na}^{\mathrm{co}}_{\mathrm{LB}}(\mathtt{a}), \mathrm{na}^{\mathrm{co}}_{\mathrm{UB}}(\mathtt{a})]$, $\mathtt{a} \in \Lambda^{\mathrm{co}}$, and $\mathrm{na}^{\mathrm{nc}}_{\mathtt{a}}(G) \in [\mathrm{na}^{\mathrm{nc}}_{\mathrm{LB}}(\mathtt{a}), \mathrm{na}^{\mathrm{nc}}_{\mathrm{UB}}(\mathtt{a})]$, $\mathtt{a} \in \Lambda^{\mathrm{nc}}$; $\mathrm{ns}_{\mu}(G) \in [\mathrm{na}_{\mathrm{LB}}(\mu), \mathrm{na}_{\mathrm{UB}}(\mu)]$, $\mu \in \Lambda_{\mathrm{dg}}$, $\mathrm{ns}^{\mathrm{co}}_{\mu}(G) \in [\mathrm{ns}^{\mathrm{co}}_{\mathrm{LB}}(\mu), \mathrm{ns}^{\mathrm{co}}_{\mathrm{UB}}(\mu)]$, $\mu \in \Lambda^{\mathrm{co}}_{\mathrm{dg}}$, and $\mathrm{ns}^{\mathrm{nc}}_{\mu}(G) \in [\mathrm{ns}^{\mathrm{nc}}_{\mathrm{LB}}(\mu), \mathrm{ns}^{\mathrm{nc}}_{\mathrm{UB}}(\mu)]$, $\mu \in \Lambda^{\mathrm{nc}}_{\mathrm{dg}}$; $\mathrm{ac}^{\mathrm{co}}_{\nu}(G) \in [\mathrm{ac}^{\mathrm{co}}_{\mathrm{LB}}(\nu), \mathrm{ac}^{\mathrm{co}}_{\mathrm{UB}}(\nu)]$, $\nu \in \Gamma^{\mathrm{co}}_{\mathrm{ac}}$, $\mathrm{ac}^{\mathrm{in}}_{\nu}(G) \in [\mathrm{ac}^{\mathrm{in}}_{\mathrm{LB}}(\nu), \mathrm{ac}^{\mathrm{in}}_{\mathrm{UB}}(\nu)]$, $\nu \in \Gamma^{\mathrm{in}}_{\mathrm{ac}}$, and $\mathrm{ac}^{\mathrm{ex}}_{\nu}(G) \in [\mathrm{ac}^{\mathrm{ex}}_{\mathrm{LB}}(\nu), \mathrm{ac}^{\mathrm{ex}}_{\mathrm{UB}}(\nu)]$, $\nu \in \Gamma^{\mathrm{ex}}_{\mathrm{ac}}$; $\mathrm{ec}^{\mathrm{co}}_{\gamma}(G) \in [\mathrm{ec}^{\mathrm{co}}_{\mathrm{LB}}(\gamma), \mathrm{ec}^{\mathrm{co}}_{\mathrm{UB}}(\gamma)]$, $\gamma \in \Gamma^{\mathrm{co}}$, $\mathrm{ec}^{\mathrm{in}}_{\gamma}(G) \in [\mathrm{ec}^{\mathrm{in}}_{\mathrm{LB}}(\gamma), \mathrm{ec}^{\mathrm{in}}_{\mathrm{UB}}(\gamma)]$, $\gamma \in \Gamma^{\mathrm{in}}$, and $\mathrm{ec}^{\mathrm{ex}}_{\gamma}(G) \in [\mathrm{ec}^{\mathrm{ex}}_{\mathrm{LB}}(\gamma), \mathrm{ec}^{\mathrm{ex}}_{\mathrm{UB}}(\gamma)]$, $\gamma \in \Gamma^{\mathrm{ex}}$; and For each edge $e \in E_{(\geq 2)} \cup E_{(\geq 1)}$, $|\{e \in E(P_e) \mid \beta(e) = m\}| \in [\mathrm{bd}_{m,\mathrm{LB}}(e), \mathrm{bd}_{m,\mathrm{UB}}(e)]$.

Figure 3(d) illustrates a $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$-extension G of G_{C} in Figure 3(a) with the specifications in Examples 1 to 3.

5 An MILP Formulation for Stage 4

In this section, we show an outline of an MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ in Stage 4 for inferring a chemical ρ-lean cyclic graph $G \in \mathcal{G}(G_{\mathrm{C}}, \sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$. The details of the MILP can be found in the detailed preprint version [3].

Scheme Graph Our method first regards a given seed graph G_{C} as a digraph and then adds some more vertices and edges to construct a digraph, called a *scheme graph* $\mathrm{SG} = (\mathcal{V}, \mathcal{E})$ so that any $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$-extension H of G_{C} can be chosen as a subgraph of SG.

For a given specification $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$, define integers that determine the size of a scheme graph SG as follows. $d_{\max} := 3$ if $\mathrm{dg}^{\mathrm{nc}}_{4,\mathrm{UB}} = 0$; $d_{\max} := 4$ if $\mathrm{dg}^{\mathrm{nc}}_{4,\mathrm{UB}} \geq 1$, $t_{\mathrm{C}} := |V_{\mathrm{C}}|$, $t_{\mathrm{T}} := \mathrm{cs}_{\mathrm{UB}} - |V_{\mathrm{C}}|$ and $t_{\mathrm{F}} := n^* - \mathrm{cs}_{\mathrm{LB}}$. Let n_{C}, n_{T} and n_{F} denote the number of "edges" in the rooted tree $T(d_{\max}-2, d_{\max}-1, \rho)$, $T(2, d_{\max}-1, \rho)$ and $T(d_{\max}-1, d_{\max}-1, \rho)$, respectively.

Formally the scheme graph $\mathrm{SG} = (\mathcal{V}, \mathcal{E})$ is defined with a vertex set $\mathcal{V} = V_{\mathrm{C}} \cup V_{\mathrm{T}} \cup V_{\mathrm{F}} \cup V^{\mathrm{ex}}_{\mathrm{C}} \cup V^{\mathrm{ex}}_{\mathrm{T}} \cup V^{\mathrm{ex}}_{\mathrm{F}}$ and an edge set $\mathcal{E} = E_{\mathrm{C}} \cup E_{\mathrm{T}} \cup E_{\mathrm{F}} \cup E_{\mathrm{CT}} \cup E_{\mathrm{TC}} \cup E_{\mathrm{CF}} \cup E_{\mathrm{TF}} \cup E^{\mathrm{ex}}_{\mathrm{C}} \cup E^{\mathrm{ex}}_{\mathrm{T}} \cup E^{\mathrm{ex}}_{\mathrm{F}}$ that consist of the following sets.

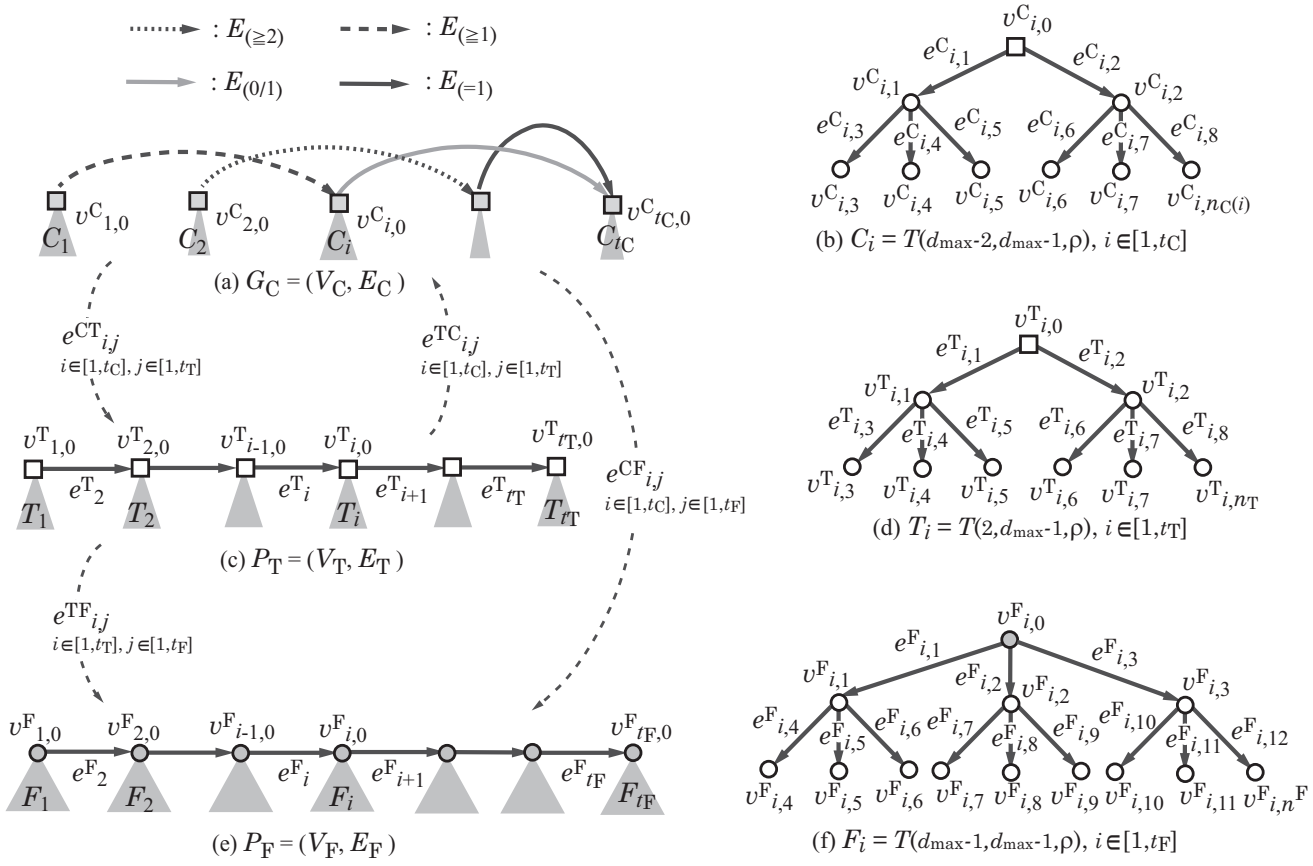


Figure 4: An illustration of a scheme graph SG: (a) A seed graph G_{C}; (b) A tree C_i, $i \in [1, t_{\mathrm{C}}]$ rooted at a core-vertex $v^{\mathrm{C}}{}_{i,0} \in V_{\mathrm{C}}$; (c) A path P_{T} of length $t_{\mathrm{T}} - 1$; (d) A tree T_i, $i \in [1, t_{\mathrm{T}}]$ rooted at a core-vertex $v^{\mathrm{T}}{}_{i,0} \in V_{\mathrm{T}}$; (e) A path P_{F} of length $t_{\mathrm{F}} - 1$; (f) A rooted tree F_i, $i \in [1, t_{\mathrm{F}}]$ rooted at a ρ-internal vertex $v^{\mathrm{F}}{}_{i,0} \in V_{\mathrm{F}}$.

Construction of the core $\mathrm{Cr}(H)$ **of a** $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$**-extension** H **of** G_{C}**:** Denote the vertex set V_{C} and the edge set E_{C} in the seed graph G_{C} by $V_{\mathrm{C}} = \{v^{\mathrm{C}}{}_{i,0} \mid i \in [1, t_{\mathrm{C}}]\}$ and $E_{\mathrm{C}} = \{a_i \mid i \in [1, m_{\mathrm{C}}]\}$, respectively, where V_{C} is always included in $\mathrm{Cr}(H)$. For including additional core-vertices in $\mathrm{Cr}(H)$, introduce a path $P_{\mathrm{T}} = (V_{\mathrm{T}} = \{v^{\mathrm{T}}{}_{1,0}, v^{\mathrm{T}}{}_{2,0}, \ldots, v^{\mathrm{T}}{}_{t_{\mathrm{T}},0}\}, E_{\mathrm{T}} = \{e^{\mathrm{T}}{}_1, e^{\mathrm{T}}{}_2, \ldots, e^{\mathrm{T}}{}_{t_{\mathrm{T}}}\})$ of length $t_{\mathrm{T}} - 1$ and a set E_{CT} (resp., E_{TC}) of directed edges $e^{\mathrm{CT}}{}_{i,j} = (v^{\mathrm{C}}{}_{i,0}, v^{\mathrm{T}}{}_{j,0})$ (resp., $e^{\mathrm{TC}}{}_{i,j} = (v^{\mathrm{T}}{}_{j,0}, v^{\mathrm{C}}{}_{i,0})$) $i \in [1, t_{\mathrm{C}}]$, $j \in [1, t_{\mathrm{T}}]$. In $\mathrm{Cr}(H)$, an edge $a_k = (v^{\mathrm{C}}{}_{i,0}, v^{\mathrm{C}}{}_{i',0}) \in E_{(\geq 2)} \cup E_{(\geq 1)}$ is allowed to be replaced with a path P_k from core-vertex $v^{\mathrm{C}}{}_{i,0}$ to core-vertex $v^{\mathrm{C}}{}_{i',0}$ that visits a set of consecutive vertices $v^{\mathrm{T}}{}_{j,0}, v^{\mathrm{T}}{}_{j+1,0}, \ldots, v^{\mathrm{T}}{}_{j+p,0} \in V_{\mathrm{T}}$ and edge $e^{\mathrm{TC}}{}_{i,j} = (v^{\mathrm{C}}{}_{i,0}, v^{\mathrm{T}}{}_{j,0}) \in E_{\mathrm{CT}}$, then edges $e^{\mathrm{T}}{}_{j+1}, e^{\mathrm{T}}{}_{j+2}, \ldots, e^{\mathrm{T}}{}_{j+p} \in E_{\mathrm{T}}$ and finally edge $e^{\mathrm{TC}}{}_{i',j+p} = (v^{\mathrm{T}}{}_{j+p,0}, v^{\mathrm{C}}{}_{i',0}) \in E_{\mathrm{TC}}$. The vertices in V_{T} in the path will be core-vertices in $\mathrm{Cr}(H)$.

Construction of paths with ρ**-internal edges in a** $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$**-extension** H **of** G_{C}**:** Introduce a path $P_{\mathrm{F}} = (V_{\mathrm{F}} = \{v^{\mathrm{F}}{}_{1,0}, v^{\mathrm{F}}{}_{2,0}, \ldots, v^{\mathrm{F}}{}_{t_{\mathrm{F}},0}\}$, $E_{\mathrm{F}} = \{e^{\mathrm{F}}{}_1, e^{\mathrm{F}}{}_2, \ldots, e^{\mathrm{F}}{}_{t_{\mathrm{F}}}\})$ of length $t_{\mathrm{F}} - 1$, a set E_{CF} of directed edges $e^{\mathrm{CF}}{}_{i,j} = (v^{\mathrm{C}}{}_{i,0}, v^{\mathrm{F}}{}_{j,0})$, $i \in [1, t_{\mathrm{C}}]$, $j \in [1, t_{\mathrm{F}}]$, and a set E_{TF} of directed edges $e^{\mathrm{TF}}{}_{i,j} = (v^{\mathrm{T}}{}_{i,0}, v^{\mathrm{F}}{}_{j,0})$, $i \in [1, t_{\mathrm{T}}]$, $j \in [1, t_{\mathrm{F}}]$. In H, a path P with ρ-internal edges

that starts from a core-vertex ${v^{\mathrm{C}}}_{i,0} \in V_{\mathrm{C}}$ (resp., ${v^{\mathrm{T}}}_{i,0} \in V_{\mathrm{T}}$) visits a set of consecutive vertices ${v^{\mathrm{F}}}_{j,0}, {v^{\mathrm{F}}}_{j+1,0}, \ldots, {v^{\mathrm{F}}}_{j+p,0} \in V_{\mathrm{F}}$ and edge ${e^{\mathrm{CF}}}_{i,h} = ({v^{\mathrm{C}}}_{i,0}, {v^{\mathrm{F}}}_{j,0}) \in E_{\mathrm{CF}}$ (resp., ${e^{\mathrm{TF}}}_{i,j} = ({v^{\mathrm{T}}}_{i,0}, {v^{\mathrm{F}}}_{j,0}) \in E_{\mathrm{TF}}$) and edges ${e^{\mathrm{F}}}_{j+1}, {e^{\mathrm{F}}}_{j+2}, \ldots, {e^{\mathrm{F}}}_{j+p} \in E_{\mathrm{F}}$. In H, the edges and the vertices (except for ${v^{\mathrm{C}}}_{i,0}$) in the path P are regarded as ρ-internal edges and ρ-internal vertices, respectively.

Construction of ρ-fringe-trees in a $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$-extension H of G_{C}: In H, the root of a ρ-fringe-tree can be any vertex in $V_{\mathrm{C}} \cup V_{\mathrm{T}} \cup V_{\mathrm{F}}$. Let $\mathrm{X} \in \{\mathrm{C}, \mathrm{T}, \mathrm{F}\}$. Introduce a rooted tree X_i, $i \in [1, t_{\mathrm{X}}]$ at each vertex ${v^{\mathrm{X}}}_{i,0}$, where each C_i is isomorphic to $T(d_{\max} - 2, d_{\max} - 1, \rho)$, each T_i is isomorphic to $T(2, d_{\max} - 1, \rho)$ and each F_i is isomorphic to $T(d_{\max} - 1, d_{\max} - 1, \rho)$. The j-th vertex (resp., edge) in each rooted tree X_i is denoted by ${v^{\mathrm{X}}}_{i,j}$ (resp., ${e^{\mathrm{X}}}_{i,j}$) See Figure 4. Let $V_{\mathrm{X}}^{\mathrm{ex}}$ and $E_{\mathrm{X}}^{\mathrm{ex}}$ denote the set of non-root vertices ${v^{\mathrm{X}}}_{i,j}$ and the set of edges ${e^{\mathrm{X}}}_{i,j}$ over all rooted trees X_i, $i \in [1, t_{\mathrm{X}}]$. In H, a ρ-fringe-tree is selected as a subtree of X_i, $i \in [1, t_{\mathrm{X}}]$ with root ${v^{\mathrm{X}}}_{i,0}$.

An MILP for Choosing a Chemical Graph from a Scheme Graph

Let K denote the dimension of a feature vector $x = f(G)$ used in constructing a prediction function ψ over a set of chemical graphs G. Based on the scheme graph SG, we obtain the following MILP formulation.

Theorem 1. *Let $(\sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$ be a target specification and $|\Gamma| = |\Lambda_{\mathrm{dg}}^{\mathrm{co}}| + |\Lambda_{\mathrm{dg}}^{\mathrm{nc}}| + |\Gamma^{\mathrm{co}}| + |\Gamma^{\mathrm{in}}| + |\Gamma^{\mathrm{co}}|$ for sets of chemical symbols and edge-configuration in $\sigma_{\alpha\beta}$. Then there is an MILP $\mathcal{M}(x, g; \mathcal{C}_2)$ that consists of variable vectors $x \in \mathbb{R}^K$ and $g \in \mathbb{R}^q$ for an integer $q = O(\mathrm{cs}_{\mathrm{UB}}(|E_{\mathrm{C}}|+n^*)+(|E_{\mathrm{C}}|+|\mathcal{V}|)|\Gamma|)$ and a set $\mathcal{C}_2$ of $O([\mathrm{cs}_{\mathrm{UB}}(|E_{\mathrm{C}}| + n^*) + |\mathcal{V}|]|\Gamma|)$ constraints on x and g such that: (x^*, g^*) is feasible to $\mathcal{M}(x, g; \mathcal{C}_2)$ if and only if g^* forms a chemical ρ-lean graph $G \in \mathcal{G}(G_{\mathrm{C}}, \sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}}, \sigma_{\alpha\beta})$ such that $f(G) = x^*$.*

Note that our MILP requires only $O(n^*)$ variables and constraints when the branch-parameter ρ, integers $|E_{\mathrm{C}}|$, $\mathrm{cs}_{\mathrm{UB}}$ and $|\Gamma|$ are constant. We explain the basic idea of our MILP in Theorem 1. The MILP mainly consists of the following three types of constraints.

C1. Constraints for selecting a ρ-lean graph $H \in \mathcal{H}(G_{\mathrm{C}}, \sigma_{\mathrm{co}}, \sigma_{\mathrm{nc}})$ as a subgraph of the scheme graph SG;

C2. Constraints for assigning chemical elements to vertices and multiplicity to edges to determine a chemical graph $G = (H, \alpha, \beta)$; and

C3. Constraints for computing descriptors from the selected chemical graph G.

In the constraints of C1, more formally we prepare the following.
Variables:
a binary variable $v^{\mathrm{X}}(i, j) \in \{0, 1\}$ for each vertex ${v^{\mathrm{X}}}_{i,j} \in V_{\mathrm{X}}$, $\mathrm{X} \in \{\mathrm{C}, \mathrm{T}, \mathrm{F}\}$ so that $v^{\mathrm{X}}(i, j) = 1 \Leftrightarrow$ vertex ${v^{\mathrm{X}}}_{i,j}$ is used in a graph H selected from SG;
a binary variable $e^{\mathrm{X}}(i) \in \{0, 1\}$ (resp., $e^{\mathrm{C}}(i) \in \{0, 1\}$) for each edge ${e^{\mathrm{X}}}_i \in E_{\mathrm{T}} \cup E_{\mathrm{F}}$ (resp., ${e^{\mathrm{C}}}_i = a_i \in E_{(\geq 2)} \cup E_{(\geq 1)} \cup E_{(0/1)}$) so that $e^{\mathrm{X}}(i) = 1 \Leftrightarrow$ edge ${e^{\mathrm{X}}}_i$ is used in a graph H selected from SG. To save the number of variables in our MILP formulation, we do not prepare a binary variable $e^{\mathrm{X}}(i, j) \in \{0, 1\}$ for any edge ${e^{\mathrm{X}}}_{i,j} \in E_{\mathrm{CT}} \cup E_{\mathrm{TC}} \cup E_{\mathrm{CF}} \cup E_{\mathrm{TC}}$, where we represent a choice of edges in these sets by a set of $O(n^*|E_{\mathrm{C}}|)$ variables (see [3] for the details);

Constraints:
linear constraints so that each ρ-fringe-tree of a graph H from SG is selected a subtree of some of the rooted trees C_i, $i \in [1, t_{\rm C}]$, T_i, $i \in [1, t_{\rm T}]$ and F_i, $i \in [1, t_{\rm F}]$;
linear constraints such that each edge $e^{\rm C}{}_i = a_i \in E_{(=1)}$ is always used as a core-edge in H and each edge $e^{\rm C}{}_i = a_i \in E_{(0/1)}$ is used as a core-edge in H if necessary;
linear constraints such that for each edge $a_k = (v^{\rm C}{}_i, v^{\rm C}{}_{i'}) \in E_{(\geq 2)}$, vertex $v^{\rm C}{}_i \in V_{\rm C}$ is connected to vertex $v^{\rm C}{}_{i'} \in V_{\rm C}$ in H by a path P_k that passes through some core-vertices in $V_{\rm T}$ and edges $e^{\rm CT}{}_{i,j}, e^{\rm T}{}_{j+1}, e^{\rm T}{}_{j+2}, \ldots, e^{\rm T}{}_{j+p}, e^{\rm TC}{}_{i',j+p}$ for some integers j and p;
linear constraints such that for each edge $a_k = (v^{\rm C}{}_i, v^{\rm C}{}_{i'}) \in E_{(\geq 1)}$, either the edge a_k is used as a core-edge in H or vertex $v^{\rm C}{}_i \in V_{\rm C}$ is connected to vertex $v^{\rm C}{}_{i'} \in V_{\rm C}$ in H by a path P_k as in the case of edges in $E_{(\geq 2)}$;
linear constraints for selecting a path P with ρ-internal edges $e^{\rm CF}{}_{i,j}$ (or $e^{\rm TF}{}_{i,j}$), $e^{\rm F}{}_{j+1}, e^{\rm F}{}_{j+2}, \ldots, e^{\rm F}{}_{j+p}$ for some integers j and p.

In the constraints of C2, we prepare an integer variable $\alpha^{\rm X}(i,j)$ for each vertex $v^{\rm X}{}_{i,j} \in \mathcal{V}$, ${\rm X} \in \{{\rm C}, {\rm T}, {\rm F}\}$ in the scheme graph that represents the chemical element $\alpha(v^{\rm X}{}_{i,j}) \in \Lambda$ if $v^{\rm X}{}_{i,j}$ is in a selected graph H (or $\alpha(v^{\rm X}{}_{i,j}) = 0$ otherwise); integer variables $\beta^{\rm C} : E_{\rm C} \to [0,3]$, $\beta^{\rm T} : E_{\rm T} \to [0,3]$ and $\beta^{\rm F} : E_{\rm F} \to [0,3]$ that represent the bond-multiplicity of edges in $E_{\rm C} \cup E_{\rm T} \cup E_{\rm F}$; and integer variables $\beta^+, \beta^- : E_{(\geq 2)} \cup E_{(\geq 1)} \to [0,3]$ and $\beta^{\rm in} : V_{\rm C} \cup V_{\rm T} \to [0,3]$ that represent the bond-multiplicity of edges in $E_{\rm CT} \cup E_{\rm TC} \cup E_{\rm CF} \cup E_{\rm TF}$. This determines a chemical graph $G = (H, \alpha, \beta)$. Also we include constraints for a selected chemical graph G to satisfy the valence condition at each vertex v with the edge-configurations $\tau(e)$ of the edges incident to v and the chemical specification $\sigma_{\alpha\beta}$.

In the constraints of C3, we introduce a variable for each descriptor and constraints with some more variables to compute the value of each descriptor in $f(G)$ for a selected chemical graph G.

6 A New Mechanism for Stage 5

This section describes the idea of a new method for Stage 5. Execution of Stage 5; i.e. generating chemical graphs G^* that satisfy $f(G^*) = x^*$ for a given feature vector $x^* \in \mathbb{Z}_+^K$ is a challenging issue for a relatively large instance with size $n(G^*) \geq 20$. There have been proposed algorithms for Stage 5 for classes of graphs with rank 0 to 2 [9, 24, 25, 26]. All of these are designed based on the branch-and-bound method where an enormous number of chemical graphs are constructed by repeatedly appending and removing a vertex one by one until a target chemical graph is constructed. These algorithms can generate a target chemical graph with size $n(G^*) \leq 20$. To break this barrier, Azam et al. [5] recently employed the dynamic programming method for designing a new algorithm in Stage 5 based on "frequency vectors" (where the frequency vector consists of some descriptors of the feature vector and the rest of descriptors can be derived from the frequency vector). They defined the frequency vector $f(G)$ of a chemical graph G to be the occurrence of each adjacency-configuration in G. Note that a single frequency vector can represent a large number of chemical graphs. The search space over frequency vectors is much more compact than that over chemical graphs. They also observed that most of the acyclic chemical compounds in the PubChem database have at most three leaf 2-branches. Their algorithm constructs the frequency

vectors of some subtrees of such a chemical acyclic graph G^* without directly building subtrees of G^* (until a required chemical acyclic graph G^* is constructed from a final set of frequency vectors). Given a vector x, their algorithm generates chemical acyclic graphs G^* with at most three leaf 2-branches such that $f(G^*) = x$ for $n(G^*) = 50$.

However, for a class of graphs with a different rank, we may need to design again a new algorithm by the dynamic programming method. Moreover, algorithms for higher ranks can be more complicated and do not run as fast as the algorithm for acyclic graphs due to Azam et al. [5].

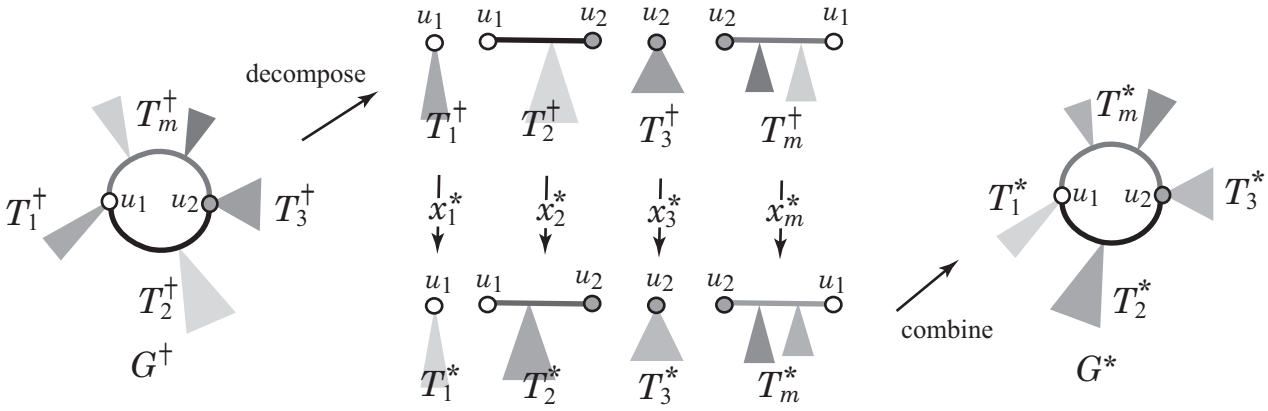


Figure 5: An illustration of a new mechanism to Stage 5, where a given chemical graph $G^\dagger$ is decomposed into chemical trees $T_i^\dagger$, $i = 1, 2, \ldots, m$ based on a set $V_B = \{u_1, u_2\}$ of core-vertices and a chemical tree T_i^* such that $f(T_i^*) = x_i^*$ is constructed for each vector $x_i^* = f(T_i^\dagger)$, before a new target graph G_1^* is obtained as a combination of $T_1^*, \ldots, T_m^*$.

In this paper, as a new mechanism of Stage 5, we adopt an idea of utilizing the chemical graph $G^\dagger \in \mathcal{G}$ obtained as part of a feasible solution of an MILP in Stage 4. The frequency vector $f(G)$ is now set to be the occurrence of each edge-configuration in G. In other words, we modify the chemical graph $G^\dagger$ to generate other chemical graphs G^* that are "chemically isomorphic" to $G^\dagger$ in the sense that $f(G^*) = f(G^\dagger)$ holds. Informally speaking, we reduce the problem of finding such a graph G^* into a problem of generating chemical acyclic graphs with two 2-leaf branches, to which we have obtained an efficient dynamic programming algorithm [5]. We first decompose $G^\dagger$ into a collection of chemical trees $T_1^\dagger, T_2^\dagger, \ldots, T_m^\dagger$ such that for a subset V_B of the core-vertices of $G^\dagger$, any tree $T_i^\dagger$ contains at most two vertices in V_B, as illustrated in Figure 5. Let x_i^* denote the feature vector $f(T_i^\dagger)$. For each index i, we generate chemical acyclic graphs T_i^* such that $f(T_i^*) = x_i^*$. Finally we combine the generated chemical trees $T_1^*, T_2^*, \ldots, T_m^*$ to construct a chemical cyclic graph G^* such that $f(G^*) = \sum_{i \in [1,m]} x_i^* = f(G^\dagger)$. (See the detailed preprint version [3] for the details on the dynamic programming algorithm.)

7 Experimental Results

We implemented our method of Stages 1 to 5 for inferring chemical 2-lean graphs and conducted experiments to evaluate the computational efficiency. We executed the experiments on a PC with

Processor: 3.0 GHz Core i7-9700 (3.0GHz) Memory: 16 GB RAM DDR4. We used ChemDoodle version 10.2.0 for constructing 2D drawings of chemical graphs.

To conduct experiments for Stages 1 to 5. we selected three chemical properties π: octanol/water partition coefficient (Kow), boiling point (Bp) and melting point (Mp).

Results on Phase 1.

We implemented Stages 1, 2 and 3 in Phase 1 as follows.

Stage 1. We set a graph class $\mathcal{G}$ to be the set of all chemical graphs with rank at least 1, and set a branch-parameter ρ to be 2. For each property $\pi \in \{\text{Kow}, \text{Bp}, \text{Mp}\}$, we first select a set Λ of chemical elements and then collect a data set D_π on chemical cyclic graphs over the set Λ of chemical elements provided by HSDB from PubChem.

Table 1 shows the size and range of data sets that we prepared for each chemical property in Stage 1, where we denote the following: Λ: the set of selected chemical elements (hydrogen atoms are added at the final stage); $|D_\pi|$: the size of data set D_π over Λ for property π; $|\Gamma^{\text{co}}|$, $|\Gamma^{\text{in}}|$, $|\Gamma^{\text{ex}}|$: the number of different edge-configurations of core-edges, 2-internal edges and 2-external edges over the compounds in D_π; $[\underline{n}, \overline{n}]$, $[\underline{\text{cs}}, \overline{\text{cs}}]$, $[\underline{\text{ch}}, \overline{\text{ch}}]$, $[\underline{\text{bl}}, \overline{\text{bl}}]$: the minimum and maximum values of $n(G)$, $\text{cs}(G)$, $\text{ch}(G)$ and $\text{bl}_2(G)$ over the compounds G in D_π; and $[\underline{a}, \overline{a}]$: the minimum and maximum values of $a(G)$ in π over compounds G in D_π.

Table 1: Data Sets for Stage 1 in Phase 1.

π	Λ	$\lvert D_\pi\rvert$	$\lvert\Gamma^{\text{co}}\rvert$	$\lvert\Gamma^{\text{in}}\rvert$	$\lvert\Gamma^{\text{ex}}\rvert$	$[\underline{n}, \overline{n}]$	$[\underline{\text{cs}}, \overline{\text{cs}}]$	$[\underline{\text{ch}}, \overline{\text{ch}}]$	$[\underline{\text{bl}}, \overline{\text{bl}}]$	$[\underline{a}, \overline{a}]$
Kow	C,O,N	424	23	19	41	[5, 58]	[3, 43]	[0, 19]	[0, 2]	[-7.53,13.45]
Kow	C,O,N,S,Cl	580	27	24	59	[5, 69]	[3, 43]	[0, 19]	[0, 5]	[-7.53,13.45]
Bp	C,O,N	175	19	13	30	[5, 30]	[3, 24]	[0, 12]	[0, 2]	[31.5,470.0]
Bp	C,O,N,S,Cl	219	20	14	39	[5, 30]	[3, 24]	[0, 12]	[0, 2]	[31.5,470.0]
Mp	C,O,N	256	22	15	36	[4, 122]	[3, 87]	[0, 28]	[0, 3]	[-142.5,300.0]
Mp	C,O,N,S,Cl	340	25	19	48	[4, 122]	[3, 87]	[0, 28]	[0, 3]	[-142.5,300.0]

Stage 2. We used a feature function f that consists of the descriptors defined in Section 2.

Stage 3. We used `scikit-learn` version 0.23.2 with Python 3.8.5, MLPRegressor and ReLU activation function to construct ANNs $\mathcal{N}$. We evaluated the resulting prediction function $\psi_{\mathcal{N}}$ with cross-validation over five subsets $D_\pi^{(i)}$, $i \in [1, 5]$ of a given data set D_π.

Table 2 shows the results on Stages 2 and 3, where we denote the following: K: the number of descriptors for the chemical compounds in data set D_π for property π; Λ: the set of selected chemical elements (hydrogen atoms are added at the final stage); Architecture: $(K, a, 1)$ (resp., $(K, a_1, a_2, 1)$) consists of an input layer with K nodes, a hidden layer with a nodes (resp., two hidden layers with a_1 and a_2 nodes, respectively) and an output layer with a single node, where K is equal to the number of descriptors in the feature vector; L-time: the average time (sec) to

Table 2: Results of Stages 2 and 3 in Phase 1.

π	Λ	Architecture	L-Time	test R^2 (ave.)	test R^2 (best)
K_{ow}	`C,O,N`	(118,13,1)	77.6	0.952	0.961
K_{ow}	`C,O,N,S,Cl`	(149,13,1)	91.0	0.932	0.948
BP	`C,O,N`	(97,11,1)	559.2	0.814	0.893
BP	`C,O,N,S,Cl`	(112,13,13,1)	206.6	0.722	0.871
MP	`C,O,N`	(108,42,10,1)	64.4	0.740	0.878
MP	`C,O,N,S,Cl`	(130,64,1)	230.8	0.800	0.859

construct ANNs for each trial; and test R^2 (ave), test R^2 (best): the average value and the largest value of coefficient of determination over the five tests.

From Table 2, we see that the execution of Stage 3 was considerably successful, where the best of test R^2 is around 0.86 to 0.96 for all three chemical properties.

Results on Phase 2.

We prepared the following instances (a)-(d) for conducting experiments of d Stages 4 and 5 in Phase 2.

(a) $I_{\rm a} = (G_{\rm C}, \sigma_{\rm co}, \sigma_{\rm nc}, \sigma_{\alpha\beta})$: The instance used in Section 4 to explain the target specification.

(b) $I^i_{\rm b} = (G^i_{\rm C}, \sigma^i_{\rm co}, \sigma^i_{\rm nc}, \sigma^i_{\alpha\beta})$, $i = 1, 2, 3, 4$: An instance for inferring chemical graphs with rank at most 2. Instance $I^1_{\rm b}$ is given by the monocyclic seed graph $G^1_{\rm C}$ in Figure 6(a) and Instances $I^i_{\rm b}$, $i = 1, 2, 3$ are given by the rank-2 seed graph $G^i_{\rm C}$, $i = 1, 2, 3$ in Figure 6(b)-(d). See [3] for the details of the specification $(\sigma^i_{\rm co}, \sigma^i_{\rm nc}, \sigma^i_{\alpha\beta})$.

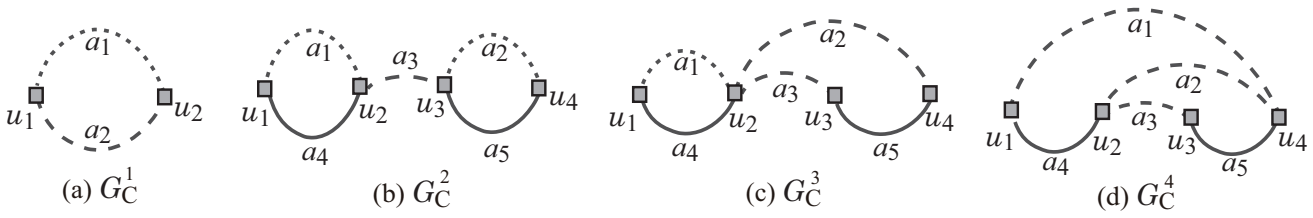


Figure 6: An illustration of seed graphs: (a) A monocyclic graph $G^1_{\rm C}$; (b) A rank-2 cyclic graph $G^2_{\rm C}$ with two vertex-disjoint cycles; (c) A rank-2 cyclic graph $G^3_{\rm C}$ with two disjoint cycles sharing a vertex; (d) A rank-2 cyclic graph $G^4_{\rm C}$ with three cycles.

We define instances in (c) and (d) in order to find chemical graphs that have an intermediate structure of given two chemical 2-lean cyclic graphs $G_A = (H_A = (V_A, E_A), \alpha_A, \beta_A)$ and $G_B = (H_B = (V_B, E_B), \alpha_B, \beta_B)$. Let $\Lambda^{\rm co}_A$ and $\Lambda^{\rm nc}_A$ denote the sets of chemical elements of the core-vertices and the non-core-vertices in G_A and $\Gamma^{\rm co}_A$, $\Gamma^{\rm in}_A$ and $\Gamma^{\rm ex}_A$ denote the sets of edge-configurations of the

core-edges, the 2-internal edges and the 2-external edges in G_A, respectively. Analogously define sets $\Lambda_B^{\rm co}$, $\Lambda_B^{\rm nc}$, $\Gamma_B^{\rm co}$, $\Gamma_B^{\rm in}$ and $\Gamma_B^{\rm ex}$ in G_B.

(c) $I_{\rm c} = (G_{\rm C}, \sigma_{\rm co}, \sigma_{\rm nc}, \sigma_{\alpha\beta})$: An instance aimed to infer a chemical graph $G^\dagger$ such that the core of $G^\dagger$ is equal to the core of G_A and the frequency of each edge-configuration in the core of $G^\dagger$ is equal to that of G_B. We use chemical compounds CID 24822711 and CID 59170444 in Figure 7(a) and (b) for G_A and G_B, respectively. Set a seed graph $G_{\rm C} = (V_{\rm C}, E_{\rm C} = E_{(=1)})$ to be the core C_A of G_A; Set $n_{\rm LB} := n^* := {\rm cs}(G_A) + (n(G_B) - {\rm cs}(G_A))$, $\Lambda^{\rm co} := \Lambda_A^{\rm co}$, $\Lambda^{\rm nc} := \Lambda_B^{\rm nc}$ and $\Lambda^*(v) := \{\alpha_A(v)\}$, $v \in V_{\rm C}$; Set ${\rm ec}_{\rm LB}^{\rm co}(\gamma) = {\rm ec}_{\rm UB}^{\rm co}(\gamma)$ to be the number of core-edges with $\gamma \in \Gamma^{\rm co}$ in G_A and ${\rm ec}_{\rm LB}^{\rm in}(\gamma) = {\rm ec}_{\rm UB}^{\rm in}(\gamma)$ (resp., ${\rm ec}_{\rm LB}^{\rm ex} = {\rm ec}_{\rm UB}^{\rm ex}(\gamma)$) to be the number of 2-internal edges (resp., 2-external edges) in G_B with edge-configuration γ.

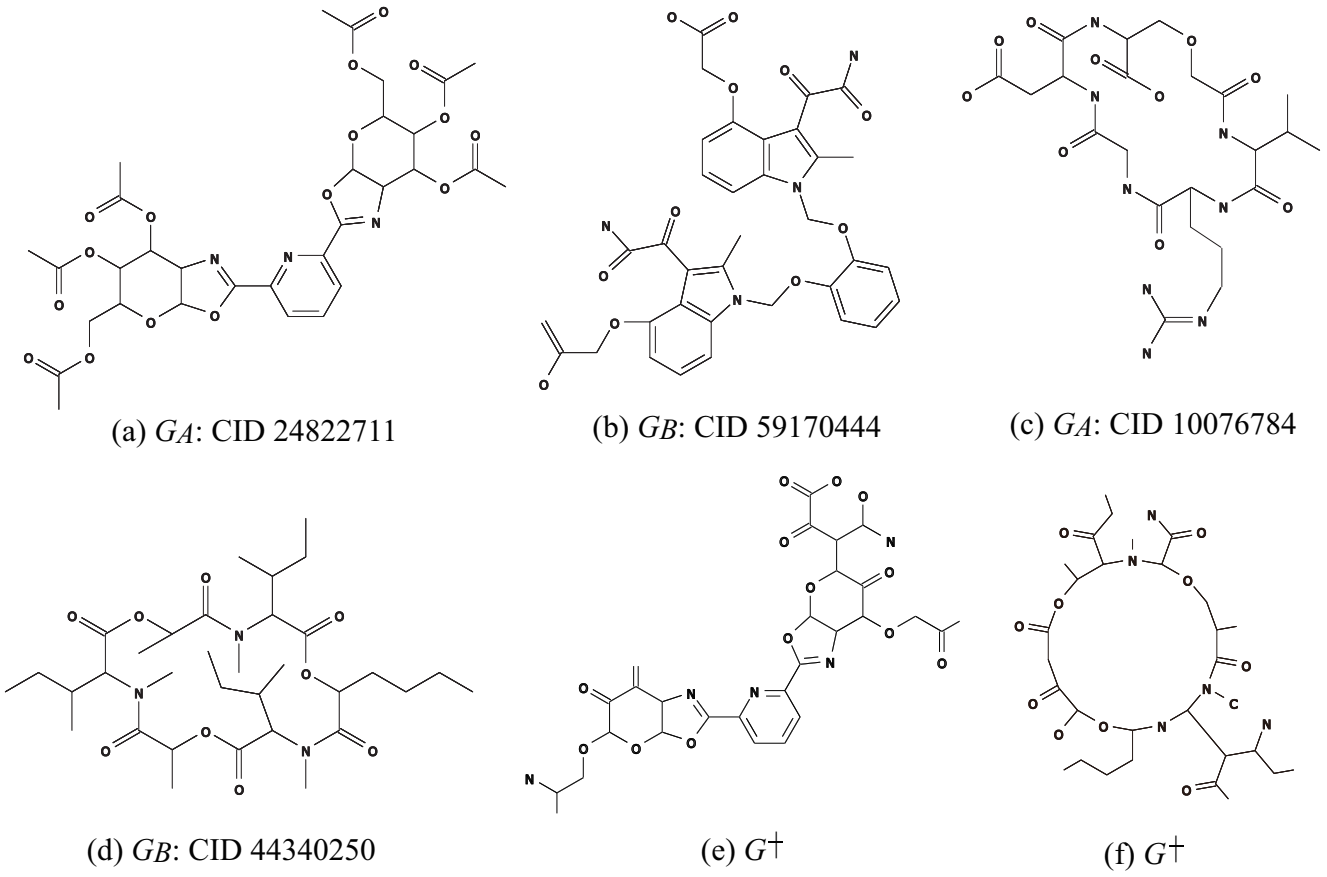
(d) $I_{\rm d} = (G_{\rm C}^1, \sigma_{\rm co}, \sigma_{\rm nc}, \sigma_{\alpha\beta})$: An instance aimed to infer a chemical monocyclic graph $G^\dagger$ such that the frequency vector of edge-configurations in $G^\dagger$ is a vector obtained by merging those of G_A and G_B. We use chemical monocyclic compounds CID 10076784 and CID 44340250 in Figure 7(c) and (d) for G_A and G_B, respectively. Set a seed graph to be the monocyclic seed graph $G_{\rm C}^1$ in Figure 6(a); Set $n_{\rm LB} := \min\{n(G_A), n(G_B)\}$; $n^* := \max\{n(G_A), n(G_B)\}$, $\Lambda^{\rm co} := \Lambda_A^{\rm co} \cup \Lambda_B^{\rm co}$ and $\Lambda^{\rm nc} := \Lambda_A^{\rm nc} \cup \Lambda_B^{\rm nc}$; For each edge-configuration $\gamma \in \Gamma^{\rm co}$ (resp., $\gamma \in \Gamma^{\rm in}, \Gamma^{\rm ex}$), let $\boldsymbol{x}_A^*(\gamma^{\rm co})$ (resp., $\boldsymbol{x}_A^*(\gamma^{\rm in}), \boldsymbol{x}_A^*(\gamma^{\rm ex})$) denote the number of core-edges (resp., 2-internal edges and 2-external edges) with γ; Analogously define $\boldsymbol{x}_B^*(\gamma^{\rm t})$, ${\rm t} \in \{{\rm co}, {\rm in}, {\rm ex}\}$; For each edge-configuration $\gamma^{\rm t} \in \Gamma^{\rm t}$, ${\rm t} \in \{{\rm co}, {\rm in}, {\rm ex}\}$, let $\boldsymbol{x}_{\rm min}^*(\gamma^{\rm t}) := \min\{\boldsymbol{x}_A^*(\gamma^{\rm t}), \boldsymbol{x}_B^*(\gamma^{\rm t})\}$ and $\boldsymbol{x}_{\rm max}^*(\gamma^{\rm t}) := \max\{\boldsymbol{x}_A^*(\gamma^{\rm t}), \boldsymbol{x}_B^*(\gamma^{\rm t})\}$; Set ${\rm ec}_{\rm LB}^{\rm t}(\gamma) := \lfloor (3/4)\boldsymbol{x}_{\rm min}^*(\gamma^{\rm t}) + (1/4)\boldsymbol{x}_{\rm max}^*(\gamma^{\rm t}) \rfloor$, ${\rm ec}_{\rm UB}^{\rm t}(\gamma) := \lceil (1/4)\boldsymbol{x}_{\rm min}^*(\gamma^{\rm t}) + (3/4)\boldsymbol{x}_{\rm max}^*(\gamma^{\rm t}) \rceil$, $\gamma^{\rm t} \in \Gamma^{\rm t}$, ${\rm t} \in \{{\rm co}, {\rm in}, {\rm ex}\}$.

Table 3 shows the features of the seven test instances, where we denote the following: Λ: the set of non-hydrogen chemical elements for inferring a target graph; $|\Gamma^{\rm co}|$, $|\Gamma^{\rm in}|$, $|\Gamma^{\rm ex}|$: the number of different edge-configurations of core-edges, 2-external edges and 2-external edges for inferring a target graph; and $[n_{\rm LB}, n^*]$, $[{\rm cs}_{\rm LB}, {\rm cs}_{\rm UB}]$, $[{\rm ch}_{\rm LB}, {\rm ch}_{\rm UB}]$, $[{\rm bl}_{\rm LB}, {\rm bl}_{\rm UB}]$: the lower and upper bounds on $n(G^\dagger)$, ${\rm cs}(G^\dagger)$, ${\rm ch}(G^\dagger)$ and ${\rm bl}_2(G^\dagger)$ for inferring a target graph $G^\dagger$.

Table 3: Features of test instances.

instance	Λ	$\lvert\Gamma^{\rm co}\rvert$	$\lvert\Gamma^{\rm in}\rvert$	$\lvert\Gamma^{\rm ex}\rvert$	$[n_{\rm LB}, n^*]$	$[{\rm cs}_{\rm LB}, {\rm cs}_{\rm UB}]$	$[{\rm ch}_{\rm LB}, {\rm ch}_{\rm UB}]$	$[{\rm bl}_{\rm LB}, {\rm bl}_{\rm UB}]$
$I_{\rm a}$	C,O,N	10	5	10	[30,50]	[20,28]	[4,6]	[2,10]
$I_{\rm b}^1$	C,O,N	28	46	74	[38,38]	[6,6]	[1,5]	[0,22]
$I_{\rm b}^2$	C,O,N	28	46	74	[50,50]	[30,30]	[1,5]	[0,34]
$I_{\rm b}^3$	C,O,N	28	46	74	[50,50]	[30,30]	[1,5]	[0,34]
$I_{\rm b}^4$	C,O,N	28	46	74	[50,50]	[30,30]	[1,5]	[0,34]
$I_{\rm c}$	C,O,N	8	3	7	[46,46]	[24,24]	[0,4]	[0,24]
$I_{\rm d}$	C,O,N	7	4	11	[40,45]	[18,18]	[0,5]	[0,32]

Tables 4 to 6 show the results on Stage 5, where we denote the following: y^*: a target value in $[\underline{a}, \overline{a}]$ for a property π; #v (resp., #c): the number of variables (resp., constraints) in the MILP



(a) G_A: CID 24822711 (b) G_B: CID 59170444 (c) G_A: CID 10076784

(d) G_B: CID 44340250 (e) $G^\dagger$ (f) $G^\dagger$

Figure 7: An illustration of chemical compounds: (a) G_A: CID 24822711; (b) G_B: CID 59170444; (c) G_A: CID 10076784; (d) G_B: CID 44340250; (e) $G^\dagger$ inferred from I_c with $y^* = 0.82$ of Kow; (f) $G^\dagger$ inferred from I_d with $y^* = 220$ of Bp.

$\mathcal{M}(x, y, g; \mathcal{C}_1, \mathcal{C}_2)$ in Stage 4; IP-time: the time (sec.) to solve the MILP in Stage 4; n, cs, ch, bl: $n(G^\dagger)$, cs$(G^\dagger)$, ch$(G^\dagger)$ and bl$_2(G^\dagger)$ in the chemical 2-lean cyclic graph $G^\dagger$ inferred in Stage 4; DP-time: the running time (sec.) to execute the dynamic programming algorithm in Stage 5; G-LB: a lower bound on the number of all chemical isomers G^* of $G^\dagger$; and #G: the number of all (or up to 100) chemical isomers G^* of $G^\dagger$ generated in Stage 5.

Figure 7(e) illustrates the chemical graph $G^\dagger$ inferred from instance I_c with $y^* = 0.82$ of Kow in Table 4.

Figure 7(f) illustrates the chemical graph $G^\dagger$ inferred from instance I_d with $y^* = 220$ of Bp in Table 5.

Recall that $I_{\rm b}^1$ (resp., $I_{\rm b}^i$, $i = 2, 3, 4$) is an instance for inferring a chemical monocyclic graph (resp., a chemical rank-2 graph) $G^\dagger$ where the sizes $n(G)$, cs(G) and ch(G) are specified. Ito et al. [12] and Zhu et al. [29] conducted similar experiments for inferring chemical monocyclic and rank-2 chemical graphs $G^\dagger$, respectively, where their feature vector contains adjacency-configuration as a descriptor, which is simpler than edge-configuration used as a descriptor in our feature vector. They solved instances with up to $n(G^\dagger) = 30$. From Tables 4 to 6, we observed that our MILP formulation successfully inferred monocyclic and rank-2 chemical graphs $G^\dagger$ with up to $n(G^\dagger) = 50$

Table 4: Results of Stages 4 and 5 for Kow.

instance	y^*	#v	#c	IP-time	n	cs	ch	bl	DP-time	G-LB	#G
I_{a}	3.00	13801	11356	44.1	47	22	5	3	0.070	2	2
I_{b}^1	2.80	43176	11202	561.2	38	6	5	4	0.112	8	2
I_{b}^2	2.80	50565	16236	1523.4	50	30	2	0	0.127	3.2×10^6	100
I_{b}^3	2.80	50634	16249	1214.1	50	30	2	0	0.199	1.0×10^5	100
I_{b}^4	2.80	50703	16260	1143.9	50	30	2	0	1.940	7.6×10^7	100
I_{c}	0.82	10348	9746	19.8	46	24	4	3	0.129	7	6
I_{d}	0.50	13858	11259	345.3	41	18	4	3	0.179	1.8×10^8	100

Table 5: Results of Stages 4 and 5 for Bp.

instance	y^*	#v	#c	IP-time	n	cs	ch	bl	DP-time	G-LB	#G
I_{a}	682	13780	11293	27.1	43	23	4	3	0.069	8	4
I_{b}^1	220	43155	11139	648.4	38	6	4	1	0.109	108	7
I_{b}^2	220	50544	16173	12058.9	50	30	2	0	0.137	1296	48
I_{b}^3	220	50613	16186	7206.3	50	30	2	0	0.169	1.5×10^7	100
I_{b}^4	220	50682	16197	4981.0	50	30	4	1	0.008	6.0×10^4	100
I_{c}	630	10327	9683	2.39	46	24	4	4	0.067	6	2
I_{d}	220	13837	11196	121.8	45	18	4	3	0.551	5.4×10^8	100

Table 6: Results of Stages 4 and 5 for Mp.

instance	y^*	#v	#c	IP-time	n	cs	ch	bl	DP-time	G-LB	#G
I_{a}	284	13699	11119	10.7	49	23	4	4	0.176	12	10
I_{b}^1	40	43074	10965	57.2	38	6	5	6	0.173	1200	40
I_{b}^2	40	50463	15999	168.8	50	30	5	4	0.237	9.5×10^5	100
I_{b}^3	40	50532	16012	149.2	50	30	3	2	0.349	2.5×10^8	100
I_{b}^4	40	50601	16023	61.5	50	30	2	0	1.730	2.0×10^6	100
I_{c}	270	10246	9509	1.71	46	24	4	3	0.065	10	6
I_{d}	240	13756	11022	27.9	44	18	4	4	0.753	2.3×10^7	100

even for our new feature vector containing edge-configuration.

8 Concluding Remarks

In this paper, we employed the new mechanism of utilizing a target chemical graph $G^\dagger$ obtained in Stage 4 of the framework for inverse QSAR/QSPR to generate a larger number of target graphs G^* in Stage 5. We showed that a family of graphs G^* that are chemically isomorphic to $G^\dagger$ can be obtained by the dynamic programming algorithm (see [3] for the details). Based on the new mechanism of Stage 5, we proposed a target specification on a seed graph as a flexible way of specifying a family of target chemical graphs. With this specification, we can realize requirements on partial topological substructure of the core of graphs and partial assignment of chemical elements and bond-multiplicity within the framework for inverse QSAR/QSPR by ANNs and MILPs.

We have implemented the proposed method to construct a system for inferring chemical compounds with a prescribed topological substructure. The results of computational experiments using such chemical properties as octanol/water partition coefficient, boiling point and melting point suggest that the proposed system can infer chemical graphs with 50 non-hydrogen atoms.

The current topological specification proposed in this paper does not allow to fix part of the non-core structure of a graph. We remark that it is not technically difficult to extend the MILP formulation in Section 5 so that a more general specification for such a case can be handled.

Authors' contributions CRediT authorship contribution statement

Jianshen Zhu: Software, Investigation. Naveed Ahmed Azam: Software, Investigation. Fan Zhang: Software, Investigation. Aleksandar Shurbevski: Software, Data Resources, Writing–review & editing. Kazuya Haraguchi: Software, Investigation. Liang Zhao: Software, Data Resources. Hiroshi Nagamochi: Conceptualization, Methodology, Formal Analysis, Writing–original draft preparation, Project administration. Tatsuya Akutsu: Conceptualization, Writing - review & editing Funding acquisition.

Acknowledgements This research was supported, in part, by Japan Society for the Promotion of Science, Japan, under Grant #18H04113.

References

[1] T. Akutsu, D. Fukagawa, J. Jansson, K. Sadakane, Inferring a graph from path frequency, Discrete Applied Mathematics 160 (2012) 1416–1428.

[2] T. Akutsu, H. Nagamochi, A mixed integer linear programming formulation to artificial neural networks, Proc. 2nd Int. Conf. Information Science and Systems, Tokyo, Japan, 2019, pp. 215–220.

[3] T. Akutsu, H. Nagamochi, A novel method for inference of chemical compounds with prescribed topological substructures based on integer programming, arXiv: 2010.09203, 2020.

[4] N. A. Azam, R. Chiewvanichakorn, F. Zhang, A. Shurbevski, H. Nagamochi, T. Akutsu, A method for the inverse QSAR/QSPR based on artificial neural networks and mixed integer linear programming, Proc. 14th Int. Conf. Biomedical Engineering Systems and Technologies, Malta, 2020, pp.101–108.

[5] N. A. Azam, J. Zhu, Y. Sun, Y. Shi, A. Shurbevski, L. Zhao, H. Nagamochi, T. Akutsu, A novel method for inference of acyclic chemical compounds with bounded branch-height based on artificial neural networks and integer programming, arXiv:2009.09646, 2020.

[6] R. S. Bohacek, C. McMartin, W. C. Guida, The art and practice of structure-based drug design: A molecular modeling perspective, Medicinal Research Reviews 16 (1996) 3–50.

[7] R. Chiewvanichakorn, C. Wang, Z. Zhang, A. Shurbevski, H. Nagamochi, T. Akutsu, A method for the inverse QSAR/QSPR based on artificial neural networks and mixed integer linear programming, 2020. Proc. 10th Int. Conf. Bioscience, Biochemistry and Bioinformatics, New York, NY, USA, 2020, 40–46.

[8] N. De Cao, T. Kipf, MolGAN: An implicit generative model for small molecular graphs, arXiv:1805.11973, 2018.

[9] H. Fujiwara, J. Wang, L. Zhao, H. Nagamochi, T. Akutsu, Enumerating treelike chemical graphs with given path frequency, Journal of Chemical Information and Modeling 48 (2008) 1345–1357.

[10] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS Central Science 4 (2018) 268–276.

[11] H. Ikebata, K. Hongo, T. Isomura, R. Maezono, R. Yoshida, Bayesian molecular design with a chemical language model, Journal of Computer-aided Molecular Design 31 (2017) 379–391.

[12] R. Ito, N. A. Azam, C. Wang, A. Shurbevski, H. Nagamochi, T. Akutsu, A novel method for the inverse QSAR/QSPR to monocyclic chemical compounds based on artificial neural networks and integer programming, Proc. 21st Int. Conf. Bioinformatics and Computational Biology, Las Vegas, Nevada, USA, 27-30 July 2020.

[13] A. Kerber, R. Laue, T. Grüner, M. Meringer, MOLGEN 4.0, Match Communications in Mathematical and in Computer Chemistry 37 (1998) 205–208.

[14] M. J. Kusner, B. Paige, J. M. Hernández-Lobato, Grammar variational autoencoder, Proc. 34th Inte. Conf. Machine Learning-Volume 70, 2017, pp. 1945–1954.

[15] J. Li, H. Nagamochi, T. Akutsu, Enumerating substituted benzene isomers of tree-like chemical graphs, IEEE/ACM Transactions on Computational Biology and Bioinformatics 15 (2016) 633–646.

[16] K. Madhawa, K. Ishiguro, K. Nakago, M. Abe, GraphNVP: an invertible flow model for generating molecular graphs, arXiv:1905.11600, 2019.

[17] T. Miyao, H. Kaneko, K. Funatsu, Inverse QSPR/QSAR analysis for chemical structure generation (from y to x), Journal of Chemical Information and Modeling 56 (2016) 286–299.

[18] H. Nagamochi, A detachment algorithm for inferring a graph from path frequency, Algorithmica 53 (2009) 207–224.

[19] J.-L. Reymond, The chemical space project, Accounts of Chemical Research 48 (2015) 722–730.

[20] C. Rupakheti, A. Virshup, W. Yang, D. N. Beratan, Strategy to discover diverse optimal molecules in the small molecule universe, Journal of Chemical Information and Modeling 55 (2015) 529–537.

[21] M. H. S. Segler, T. Kogej, C. Tyrchan, M. P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, ACS Central Science 4 (2017) 120–131.

[22] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, J. Tang, GraphAF: a flow-based autoregressive model for molecular graph generation, arXiv:2001.09382, 2020.

[23] M. I. Skvortsova, I. I. Baskin, O. L. Slovokhotova, V. A. Palyulin, N. S. Zefirov, Inverse problem in QSAR/QSPR studies for the case of topological indices characterizing molecular shape (Kier indices), Journal of Chemical Information and Computer Sciences 33 (1993) 630–634.

[24] M. Suzuki, H. Nagamochi, T. Akutsu, Efficient enumeration of monocyclic chemical graphs with given path frequencies, Journal of Cheminformatics 6 (2014) 31.

[25] Y. Tamura, Y. Y. Nishiyama, C. Wang, Y. Sun, A. Shurbevski, H. Nagamochi, T. Akutsu, Enumerating chemical graphs with mono-block 2-augmented tree structure from given upper and lower bounds on path frequencies, arXiv:2004.06367, 2020.

[26] K. Yamashita, R. Masui, X. Zhou, C. Wang, A. Shurbevski, H. Nagamochi, T. Akutsu, Enumerating chemical graphs with two disjoint cycles satisfying given path frequency specifications, arXiv:2004.08381, 2020.

[27] X. Yang, J. Zhang, K. Yoshizoe, K. Terayama, K. Tsuda, ChemTS: an efficient python library for de novo molecular generation, Science and Technology of Advanced Materials 18 (2017) 972–976.

[28] F. Zhang, J. Zhu, R. Chiewvanichakorn, A. Shurbevski, H. Nagamochi, T. Akutsu, A new integer linear programming formulation to the inverse QSAR/QSPR for acyclic chemical compounds using skeleton trees, Proc. 33rd Int. Conf. Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer LNCS 12144, 2020, pp. 433–444.

[29] J. Zhu, C. Wang A. Shurbevski, H. Nagamochi, T. Akutsu, A novel method for inference of chemical compounds of cycle index two with desired properties based on artificial neural networks and integer programming, Algorithms 13 (2020) 124.