Pseudo-codes for Graph Search Algorithm

For each base vertex v or edge e, we are given a set $\mathcal{F}$ of fringe trees. For each $T \in \mathcal{F}$,
- identify the root information such as root label, multiplicity and degree. Let $\mathtt{a}, m, d$ be label, multiplicity and degree of the root of the tree T.
- obtain $T[+\Delta]$ and vector $\boldsymbol{w}$ of $T[+\Delta]$ for each $\Delta \in [1, \mathrm{val}(\mathtt{a}) - d]$ and include $\boldsymbol{w}$ in the respective sets $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*), V^{(0)}_{\mathrm{inl}}(a, d, m; \boldsymbol{x}^*), \mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(a, d, m, h; \boldsymbol{x}^*)$, $h \leq 2$, $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta+1)}(a, d, m, p; \boldsymbol{x}^*)$

1 Computing Frequency Vectors of End-Subtrees

For an integer $h \geq 1$, element $\mathtt{a} \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, and $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ we give a procedure to compute the set $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$.

ComputeEndSubtreeOne($\mathtt{a}, d, m, h$)

Input: Element $\mathtt{a} \in \Lambda$, integer $d \in [1, \mathrm{val}(a) - 1]$, $m \in [d, \mathrm{val}(a) - 1]$, $h \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{int}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$
a non-negative integer b, the collection
$\mathcal{V}_{\mathrm{inl}}$ vector sets $\mathrm{V}_{\mathrm{inl}}(\mathtt{a}, d-1, m_{\mathtt{a}}; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a}) - 2]$
$\mathcal{V}^{(h-1)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(h-1)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$. */
Output: The set $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$, where we store these vectors in a trie.

```
W := ∅;
for each triplet (b, d_b, m_b) do
  for each triplet (a, d − 1, m_a) do
    for each y^b = (y^b_int, y^b_ex, 0) ∈ V^(h−1)_end(b, d_b, m_b; x*) do
      for each m' ∈ [1, 3] such that
          - γ^int = (a{d + 1}, b{d_b + 1}, m') ∈ Γ^int and
          - m_a + m' = m, m_a + m' + 1 ≤ val(a) and m' + m_b ≤ val(b) do
        for each y^a = (y^a_int, y^a_ex, 0) ∈ V^(0)_inl(a, d − 1, m_a; x*) do
          y_int := y^a_int + y^b_int + 1_{γ^int};
          y_ex := y^a_ex + y^b_ex; y := (y_int, y_ex, 0);
          if y ≤ x* then
            W := W ∪ {y};
          end if
        end for
      end for
    end for
  end for
end for;
Output W as V^(h)_end(a, d, m; x*).
```

2 Generating Frequency Vectors of Rooted Core-subtrees

For an integer $h \geq 1$, element $\mathtt{a} \in \Lambda$, integers $\Delta \in [2,3]$, $d \in [1, \mathrm{val}(\mathtt{a}) - \Delta]$, and $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ we give a procedure to compute the set $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$.

ComputCoreSubtreeOne$(\mathtt{a}, d, m, h)$

Input: Element $\mathtt{a} \in \Lambda$, integer $d \in [1, \mathrm{val}(a) - \Delta]$, $m \in [d, \mathrm{val}(a) - 1]$, $h \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{int}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$,
a non-negative integer b, the collection
$\mathcal{V}^{(0)}_{\mathrm{co}+\Delta+1}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d-1, m_{\mathtt{a}}, p; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a}) - \Delta - 1]$,
$p \in [0, 2(=\rho)]$
$\mathcal{V}^{(h-2-1)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(h-2-1)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$, integer $g \geq 1$. */
Output: The set $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$, where we store vectors $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$,
in a trie.

```
W := ∅;
for each triplet (b, d_b, m_b) do
  for each triplet (a, d − 1, m_a, p) do
    for each y^b = (y^b_int, y^b_ex, 0) ∈ V^(h−2−1)_end(b, d_b, m_b; x*) do
      for each m' ∈ [1, 3] such that
        - γ^int = (a{d + Δ}, b{d_b + 1}, m') ∈ Γ^int and
        - m_a + m' = m, m_a + m' + Δ ≤ val(a) and m' + m_b ≤ val(b) do
        for each w^a = (w^a_int, w^a_ex, 0) ∈ W^(0)_inl(a, d − 1, m_a, p; x*) do
          w_int := w^a_int + y^b_int + 1_{γ^int};
          w_ex := w^a_ex + y^b_ex; y := (y_int, y_ex, 1);
          if y ≤ x* then
            W := W ∪ {y};
          end if
        end for
      end for
    end for
  end for
end for;
Output W as V^(0)_co+Δ(a, d, m, h; x*).
```

3 Computing DAG Representation for v-Components

DAGRepresentationVertex$(\mathtt{a}_v, d_v, m_v, t, \Delta_v, \boldsymbol{x}^*)$

Input: /* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{int}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$,
a non-negative integer b,
integers t, $\Delta_v \in [2,3]$, element $\mathtt{a}_v \in \Lambda$,

integers $d_v \in [0, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $m_v \in [d_v, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$,
the collection $\mathcal{V}^{(0)}_{\mathrm{inl}}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$,
an integer $t \geq 0$,
$\mathcal{V}^{(h)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$, $1 \leq h \leq t$,
the collection $\mathcal{V}^{(0)}_{\mathrm{end}}$ of sets $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
the collection $\mathcal{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}$ of sets $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v - 1, m'', p; \boldsymbol{x}^*)$ for $p \leq 2$. */

Output: A vertex-labeled and edge-labeled DAG representation.
$F := \emptyset$;
$G := (N, A)$; $A := \emptyset$; $N := \emptyset$;
for each $\boldsymbol{w} \in \mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v - 1, m', p; \boldsymbol{x}^*)$ for each possible (m', p) **do**
  for each $\boldsymbol{y} \in \mathrm{V}^{(t)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$ for each possible $(\mathtt{a}_1, d_1, m_1)$ **do**
    if there exists $\gamma := (\mathtt{a}_v\{d_v + \Delta_v\}, \mathtt{a}_1\{d_1 + 1\}, m_v - m') \in \Gamma^{\mathrm{co}}$
      such that $\boldsymbol{y} + \boldsymbol{w} + \mathbf{1}_\gamma = \boldsymbol{x}^*$ **then**
      $N := N \cup \{(\boldsymbol{x}^*, t+1; \mathtt{a}_v, d_v, m_v)\}$;
      $N := N \cup \{(\boldsymbol{y}, t; \mathtt{a}_1, d_1, m_1)\}$;
      $A := A \cup \{a_{\boldsymbol{x}^*\boldsymbol{y}}\}$ and
      let the label of the arc $a_{\boldsymbol{x}^*\boldsymbol{y}}$ to be $(\boldsymbol{w}, m_v - m')$;
    end if
  end for
end for;
for each $\ell \in (t, \ldots, 1)$ **do**
  $G'' := (N'', A'') :=$ DAGSublayer$(\mathcal{V}^{(\ell-1)}_{\mathrm{end}}, G, \ell - 1, \mathcal{V}^{(0)}_{\mathrm{inl}})$;
    $N := N \cup N''$; $A := A_2 \cup A''$
end for;
Output G as DAG representations and the set F of feasible pairs of v-component.

DAGSublayer$(\mathcal{V}, G, \ell, \mathcal{V}')$

Input: A family $\mathcal{V}$ of set of vectors of trees with root label $\mathtt{a}_1$, degree
d_1 and multiplicity m_1, $G = (N, A)$,
a family of $\mathcal{V}'$ vector sets of fringe-trees,
ℓ (the height of the layer that we add in G at this stage).
Output: A DAG G' that is a super-graph of G.
$G' := G$;
for each $\boldsymbol{y}_1 \in \mathcal{V}$ **do**
  for each $\boldsymbol{w} \in \mathcal{V}'$ **do**
    if there exists $\gamma \in \Gamma^{\mathrm{in}}$ and some $\boldsymbol{y}_2 \in N$ such that
      $\boldsymbol{y}_i, i = 1, 2$ are feasible, i.e., $\boldsymbol{y}_1 + \boldsymbol{w} + 1_\gamma = \boldsymbol{y}_2$ **then**
      if $\boldsymbol{y}_1 \notin N$ **then** $N := N \cup \{(\boldsymbol{y}_1, \ell; \mathtt{a}_1, d_1, m_1)\}$;
      $A := A \cup \{a_{\boldsymbol{y}_2\boldsymbol{y}_1}\}$ and
        label the arc from $\boldsymbol{y}_2$ to $\boldsymbol{y}_1$ by $(\boldsymbol{w}, m)$,

where m is the bond multiplicity in γ
end if
end for
end for;
Output G' as a required DAG.

4 Enumerating Paths in DAG

EnumPaths(DAG)

Input: A rooted vertex-labeled and edge-labeled DAG $G = (N, A)$.
Output: All directed paths from sources to leaves.
We consider G a rooted DAG with a virtual root r that is adjacent with all source vertices;
We consider dsf ordering on the vertices of G starting from root with index 0 and
traverse G in left-right ordering on the children of each vertex;
$\mathcal{P} := \emptyset$;
Let $Q_i :=$ set dfs label of all children of the vertex with dfs label i, $i \in |N|$;
if $|Q_1| = 0$ **then**
$\mathcal{P} := \mathcal{P} \cup \{1\}$
else
while $Q_1 \neq \emptyset$ **do**
Let i be the smallest integer in Q_1;
Let $\boldsymbol{y}_1$, and $\boldsymbol{y}_i$ be the label of vertices with dfs label 1 and i, respectively, and
the label of arc between y_1 and $\boldsymbol{y}_i$ is $(\boldsymbol{w}, m)$
$P := ((\boldsymbol{y}_1, \boldsymbol{y}_i, \boldsymbol{w}, m))$;
$\mathcal{P}' :=$ PathRecursion$(P, i, \mathcal{P}, G)$; $Q_1 := Q_1 \setminus \{i\}$; $\mathcal{P} := \mathcal{P} \cup \mathcal{P}'$
end while
end if;
Output $\mathcal{P}$ as the required family of paths.

PathRecursion$(P, i, \mathcal{P}, G)$

Input: A DAG $G = (N, A)$ with dfs ordering, a path P,
a family of paths $\mathcal{P}$ an integer $i \in [2, |N|]$.
Output: Family of paths in G that can be extended from P.
$\mathcal{P}' := \emptyset$;
Let $Q_i :=$ set of dfs label of all children of the vertex with dfs label i;
if $|Q_i| = 0$ **then** $\mathcal{P}' := \mathcal{P}' \cup \{P\}$;
else
while $Q_i \neq \emptyset$ **do**
Let j be the smallest integer in Q_i;
Let $\boldsymbol{y}_i$, and $\boldsymbol{y}_j$ be the labels of the vertices with dfs label i and j, respectively, and
the label of arc between y_i and $\boldsymbol{y}_j$ is $(\boldsymbol{w}, m)$

$P' := P \oplus ((\boldsymbol{y}_i, \boldsymbol{y}_j, \boldsymbol{w}, m))$; /* sequence concatenation */
$\mathcal{P}'' :=$ PathRecursion$(P', j, \mathcal{P}', G)$; $Q_i := Q_i \setminus \{j\}$; $\mathcal{P}' := \mathcal{P}' \cup \mathcal{P}''$
end while
end if;
Output $\mathcal{P}'$ as the required family of extended paths.

5 A Complete Algorithm to Compute Target v-components

We briefly summarize how to use the procedures described thus far to obtain an algorithm. Our global constants are vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm int}, \boldsymbol{x}^*_{\rm ex}, b)$, a non-negative integer b, integer $\Delta_v \in [2, 3]$, element $\mathtt{a}_v \in \Lambda$,
integers $d_v \in [0, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $m_v \in [d_v, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$.

CompleteAlgorithmVertex(Global constants: $\mathtt{a}_v, d_v, m_v, \Delta_v, \boldsymbol{x}^*_v$, core height)
Let $\ell := |\Gamma^{\rm in}| + 2$;
$t :=$ core height $- 3$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta_v}(\mathtt{a}_v, d_v, m_v, h; \boldsymbol{x}^*_v)$ for a fixed $(\mathtt{a}_v, d_v, m_v, \Delta_v)$,
and for each $h \le \ell$ if $\ell \le 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 0$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v, m, h; \boldsymbol{x}^*_v)$ for a fixed $(\mathtt{a}_v, d_v, \Delta_v)$,
for each $m \in [d_v - 1, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $h \le 2$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_v)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
$m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*_v)$ for each $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - 2]$,
$m \in [d, \mathrm{val}(\mathtt{a}) - 2]$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{b}, d', m'; \boldsymbol{x}^*_v)$ for each $\mathtt{b} \in \Lambda$, $d' \in [1, \mathrm{val}(\mathtt{b}) - 1]$,
$m' \in [d', \mathrm{val}(\mathtt{b}) - 1]$, $1 \le h \le t$, if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute the DAG G representation of $\boldsymbol{x}^*_v$;
Enumerate the set $\mathcal{P}$ of paths from sources to leaves in G;
for each path P in G **do**
Let $P := ((\boldsymbol{x}^*, \boldsymbol{y}_h, \boldsymbol{w}_h, m_h), (\boldsymbol{y}_h, \boldsymbol{y}_{h-1}, \boldsymbol{w}_{h-1}, m_{h-1}), \ldots, (\boldsymbol{y}_1, \boldsymbol{y}_0, \boldsymbol{w}_0, m_0))$;
where $\boldsymbol{w}_h \in \mathrm{V}^{(\delta_1)}_{\mathrm{co}+(\Delta_v+1)}$, $\boldsymbol{w}_{h-1}, \ldots, \boldsymbol{w}_1 \in \mathrm{V}^{(0)}_{\mathrm{inl}}$, $\boldsymbol{w}'_0 \in \mathrm{V}^{(0)}_{\mathrm{end}}$, $h = t$;
Get a target v-component by using the trees corresponding to
$\boldsymbol{w}_h, \boldsymbol{w}_{h-1}, \ldots, \boldsymbol{w}_0$
Get the number of v-components obtained by the path P
$n(\boldsymbol{w}_h) \times \cdots \times n(\boldsymbol{w}_0)$, where $n(\boldsymbol{w}_h), \ldots, n(\boldsymbol{w}_0)$, are the number of trees with vector
$\boldsymbol{w}_h, \ldots, \boldsymbol{w}_0$, respectively
end for.

6 Generation of Frequency Vectors of Bi-rooted Core-subtrees

For an integer $h \in [h_1, h_2]$, elements $\mathtt{a}, \mathtt{a}^e \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $\Delta^e \in [1, \mathrm{val}(\mathtt{a}^e) - 1]$, $m^e \le \mathrm{val}(\mathtt{a}^e) - \Delta^e$, and $q \ge 1$, we give a procedure to compute the set

$\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h; \boldsymbol{x}^*)$.

$\textsc{ComputeBiRootedCoreSubtree}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h, q)$

Input: An integer $h \geq 0$, elements $\mathtt{a}, \mathtt{a}^e \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$,
$\Delta^e \in [1, \mathrm{val}(\mathtt{a}^e) - 1]$, $m^e \leq \mathrm{val}(\mathtt{a}^e) - \Delta^e$, and $q \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{int}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$,
a non-negative integer b, the collection
$\mathcal{V}^{(0)}_{\mathrm{co}+2}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{co}+2}(\mathtt{a}, d-1, m_{\mathtt{a}}, p; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a}) - \Delta - 1]$, $p \in [0, h]$,
for $q \geq 2$, $\mathcal{V}^{(q-1)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(q-1)}_{\mathrm{co}+1,\Delta^e}(\mathtt{b}, d', m', \mathtt{a}^e, 1, m^e, h'; \boldsymbol{x}^*)$,
$\mathtt{b} \in \Lambda$, $d' \in [1, \mathrm{val}(\mathtt{b}) - 1]$, $m' \in [d', \mathrm{val}(\mathtt{b}) - 1]$, $h' \in [0, h]$, integer $g \geq 1$. */
Output: The set $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h; \boldsymbol{x}^*)$, where we store these vectors in a trie.

```
W := ∅;
for each triplet (a, d − 1, m_a, p) do
  if q = 1 then
    if p = h and val(a) ≥ m_a + m^e then
      for each w^a ∈ V^(0)_{co+2}(a, d − 1, m_a, p; x*) do
        γ^int := (ad, a^e1, m^e); y := y^a + 1_{γ^int}
        if γ^int ∈ Γ^int and y ≤ x* then
          if y ∈ V then
            V := V ∪ {y}
          end if
        end if
      end for
    end if
  else /* q > 1 */
    for each triplet (b, d_b, m_b, h') do
      for each y^b ∈ V^(q−1)_{co+1,Δ^e}(b, d_b, m_b, a^e, 1, m^e, h'; x*) do
        for each m' ∈ [1, 3] such that
          - γ^int := (ad, b{d_b + 1}, m') ∈ Γ^int and
          - m_a + m' = m, m_a + m' + 1 ≤ val(a), m' + m_b ≤ val(b),
          - h = max{p, h'} and
          - y := y_a + y_b + 1_{γ^int} ≤ x* do
          if y ∈ V then
            V := V ∪ {y};
          end if
        end for
      end for
    end for
  end if
end for;
Output W as V^(q)_{co+1,Δ^e}(a, d, m, a^e, 1, m^e, h; x*).
```

7 Computing DAG Representation for e-Components

DAGRepresentationEdge($\mathtt{a}_i^e, m_i^e, \Delta_i^e, \delta_i, h_i\boldsymbol{x}^*$)

Input: /* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm int}, \boldsymbol{x}^*_{\rm ex}, b)$,
a non-negative integer b,
$\mathtt{a}_i^e \in \Lambda$, integers $\Delta_i^e \in [1, \mathrm{val}(\mathtt{a}_i^e) - 1]$, $m_i^e \leq \mathrm{val}(\mathtt{a}_i^e) - \Delta_i^e$,
the collection $\mathcal{V}^{(0)}_{\rm inl}$ vector sets $\mathrm{V}^{(0)}_{\rm inl}(\mathtt{a}, d, m; \boldsymbol{x}^*)$,
integers $\delta_i \geq 0$, $h_i \geq 1$, $i = 1, 2$,
$\mathcal{V}^{(h)}_{\rm end}$ of vector sets $\mathrm{V}^{(h)}_{\rm end}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$, $1 \leq h \leq \max\{\delta_1, \delta_2\}$,
the collection $\mathcal{V}^{(0)}_{\rm end}$ of sets $\mathrm{V}^{(0)}_{\rm end}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$,
the collection $\mathcal{V}^{(0)}_{\rm co+2}$ of sets $\mathrm{V}^{(0)}_{\rm co+2}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$ for all possible $\mathtt{a}, d, m$ and $h \leq \max\{h_1, h_2\}$,
$\mathcal{V}^{(0)}_{\rm co+(\Delta+1)}$ of sets $\mathrm{V}^{(0)}_{\rm co+(\Delta+1)}(\mathtt{a}, d-1, m'', p; \boldsymbol{x}^*)$ for $p \leq 2$,
for $2 \leq q_i \leq \delta_i, i = 1, 2$, families $\mathcal{V}^{(q_i)}_{{\rm end},i}(\mathtt{a}_i^e, m_i^e)$ of vector sets $\mathrm{V}^{(q_i)}_{{\rm co+1},\Delta_i^e}(\mathtt{a}_i, d_i, m_i, \mathtt{a}_i^e, 1, m_i^e, h_i; \boldsymbol{x}^*)$. */

Output: A set of feasible pairs $\boldsymbol{y}_i, i = 1, 2$ of length $\delta_i, i = 1, 2$, respectively,
two vertex-labeled and edge-labeled DAG representation of these feasible pairs of e-component,
and DAG representations of frequency vector of each non-core part of the e-component
with frequency vector $\boldsymbol{x}^*$.

$F := \emptyset$; /* to store feasible pairs for core part */
$G_i := (N_i, A_i)$; $A_i := \emptyset$; $N_i := \emptyset, i = 1, 2$;/* core part */
for each $(\mathtt{a}_i, d_i, m_i), i = 1, 2$
  for each $\gamma = (\mathtt{a}_1\{d_1 + 1\}, \mathtt{a}_2\{d_2 + 1\}, m) \in \Gamma^{\rm int}$ with
    $m \in [1, \min\{3, \mathrm{val}(\mathtt{a}_1) - m_1, \mathrm{val}(\mathtt{a}_2) - m_2\}]$ **do**
   Let L_1 denote the sorted list of vectors in $\mathrm{V}^{(\delta_1)}_{{\rm co+1},\Delta_1^e}(\mathtt{a}_1, d_1, m_1, \mathtt{a}_1^e, 1, m_1^e, h_1; \boldsymbol{x}^*)$;
   Construct the set $\overline{\mathrm{W}} := \{\overline{\boldsymbol{z}} \mid \boldsymbol{z} \in \mathrm{V}^{(\delta_2)}_{{\rm co+1},\Delta_2^e}(\mathtt{a}_2, d_2, m_2, \mathtt{a}_2^e, 1, m_2^e, h_2; \boldsymbol{x}^*)\}$ of the γ-complement vectors;
   Sort the vectors in $\overline{\mathrm{W}}$ to obtain a sorted list L_2;
   Merge L_1 and L_2 into a single sorted list L_γ of vectors in both lists (as a multiset);
   Trace the list L_γ and for each consecutive pair $\boldsymbol{z}^1, \boldsymbol{z}^2$ of vectors with $\boldsymbol{z}^1 = \boldsymbol{z}^2$
   $\boldsymbol{y}_1 := \boldsymbol{z}^1, \boldsymbol{y}_2 := \overline{\boldsymbol{z}^2}$ is a feasible pair;
   $N_i := N_i \cup \{(\boldsymbol{y}_i, \delta_i; \mathtt{a}_i, d_i, m_i, h_i)$;
   Let the label of the arc from $\boldsymbol{y}_1$ to $\boldsymbol{y}_2$ is $(\mathbf{0}, m)$;
   $F := F \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2; \mathbf{0}, m'; \mathtt{a}_1, d_1, m_1, h_1; \mathtt{a}_2, d_2, m_2, h_2)\}$
  end for
end for;
$\mathcal{C} := \emptyset$;
/* a set of vectors of rooted core subtrees for which we calculate DAG in second phase */
$G' := G_2$;
for each $\ell \in (\delta_2, \ldots, 1)$ **do**
  $(G'' := (N'', A''), \mathcal{D}) :=$ CoreDAGSublayer$(\mathcal{V}^{(\ell-1)}_{\rm co+1,2}, G', \ell - 1, \mathcal{V}^{(0)}_{\rm co+2}, h_2)$;
   $N_2 := N_2 \cup N''$; $A_2 := A_2 \cup A''$; $\mathcal{C} := \mathcal{C} \cup \mathcal{D}$
end for;

$G' := G_1$;
for each $\ell \in (\delta_1, \ldots, 1)$ **do**
$(G'' := (N'', A''), \mathcal{D}) :=$ CoreDAGSublayer$(\mathcal{V}^{(\ell-1)}_{\mathrm{co}+1,1}, G', \ell-1, \mathcal{V}^{(0)}_{\mathrm{co}+2}, h_1)$;
$N_1 := N_1 \cup N''$; $A_1 := A_1 \cup A''$; $\mathcal{C} := \mathcal{C} \cup \mathcal{D}$
end for;
for each $(\boldsymbol{y}, \mathtt{a}, d, m, t) \in \mathcal{C}$ **do**
$G''' := (N''', A''')$; $N''' := \{\boldsymbol{y}\}$; $A''' := \emptyset$;
for each $\ell \in (t-2, \ldots, 1)$ **do**
if $\ell = t-2$ **then**
$G^* := (N^*, A^*) :=$ DAGSublayer$(\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y}), G''', \ell-1, \mathcal{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m; \boldsymbol{y}))$,
where $\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y})$ is a family of vectors of end-subtrees under $\boldsymbol{y}$
with core height $\ell-1$,
$\mathcal{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m; \boldsymbol{y})$ is the family of sets $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m, p; \boldsymbol{y})$;
$N''' := N''' \cup N^*$; $A''' := A''' \cup A^*$;
else /* $\ell < t-2$ */
$G^* := (N^*, A^*) :=$ DAGSublayer$(\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y}), G''', \ell-1, \mathcal{V}_{\mathrm{inl}})$,
where $\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y})$ is a family of vectors of end-subtrees under $\boldsymbol{y}$
with core height $\ell-1$;
$N''' := N''' \cup N^*$; $A''' := A''' \cup A^*$;
end if
end for;
Output $(\boldsymbol{y}, G'''')$
end for;
Output $G_i, i = 1, 2$ as DAG representations and the set F.

CoreDAGSublayer$(\mathcal{V}, G, \ell, \mathcal{V}', h)$

Input: A family $\mathcal{V}'$ of vectors rooted core-subtrees with a root label $\mathtt{a}_1$,
degree d_1 and multiplicity m_1 and core height $t \le h$, $G = (N, A)$,
a family $\mathcal{V}$ of vectors of bi-rooted core subtrees with core height at most h and
ℓ (the height of the layer that we add in G at this stage).
Output: A DAG G' that is a super-graph of G, and a set of vectors of rooted core subtrees.
$G' := G$; $\mathcal{C} := \emptyset$;
for each $\boldsymbol{y}_1 \in \mathcal{V}$ **do**
for each $\boldsymbol{y}'_1 \in \mathcal{V}'$ **do**
Let the height of $\boldsymbol{y}_1$ and $\boldsymbol{y}'_1$ be t and t', respectively;
if there exists $\gamma \in \Gamma^{\mathrm{in}}$ and some $\boldsymbol{y}_2 \in N$ such that
$\boldsymbol{y}_i, i = 1, 2$ are feasible, i.e., $\boldsymbol{y}_1 + \boldsymbol{y}'_1 + 1_\gamma = \boldsymbol{y}_2$ and $\max\{t, t'\} = h$ **then**
if $\boldsymbol{y}_1 \notin N$ **then** $N := N \cup \{(\boldsymbol{y}_1, \ell; \mathtt{a}_1, d_1, m_1, t')\}$;
$A := A \cup \{a_{\boldsymbol{y}_2\boldsymbol{y}_1}\}$ and
label the arc from $\boldsymbol{y}_2$ to $\boldsymbol{y}_1$ by $(\boldsymbol{y}', m)$,
where m is the bond multiplicity in γ;
$\mathcal{C} := \mathcal{C} \cup \{(\boldsymbol{y}'_1, \mathtt{a}_1, d_1, m_1, t')\}$

end if
end for
end for;
Output G' as a required DAG and $\mathcal{C}$ the required set of rooted core subtrees.

8 A Complete Algorithm to Compute Target e-components

We briefly summarize how to use the procedures described thus far to obtain an algorithm. Our global constants are a frequency vector $\boldsymbol{x}^*_{\mathrm{e}}$ of an e-component, two fixed tuples $(\mathtt{a}^e_j, m^e_j, \Delta^e_j), j = 1, 2$ a lower bound $\mathrm{ch}_{\mathrm{LB}}(e)$ and an upper bound $\mathrm{ch}_{\mathrm{UB}}(e)$ on core height, where we take $\rho = 2$.

CompleteAlgorithmEdge(Global constants: $\mathtt{a}^e_j, m^e_j, \Delta^e_j, \boldsymbol{x}^*_e$, core height bounds)

$\Gamma^{\mathrm{in}}_e :=$ The set internal edges in $\boldsymbol{x}^*_e$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta+1)}(\mathtt{a}, d, m, h; \boldsymbol{x}^*_e)$ for each
  $\Delta \in [2, 3]$, $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - \Delta]$, $m \in [d, \mathrm{val}(\mathtt{a}) - \Delta]$, $h \in [0, \min\{2, \mathrm{ch}_{\mathrm{UB}}(e)\}]$;
Compute $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$;
Compute $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - 2]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 2]$;
Compute $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
  $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $1 \leq h \leq \min\{|\Gamma^{\mathrm{in}}_e| - 1, \mathrm{ch}_{\mathrm{UB}}(e) - 2 - 1\}$ if $\mathrm{ch}_{\mathrm{UB}}(e) > 2$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*_v)$ for each $\Delta \in [2, 3]$, $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
  $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $h \leq \min\{|\Gamma^{\mathrm{in}}_e| + 2, \mathrm{ch}_{\mathrm{UB}}(e)\}$, if $\mathrm{ch}_{\mathrm{UB}}(e) > 2$;
Compute $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e_j}(\mathtt{a}, d, m, \mathtt{a}^e_j, 1, m^e_j, h; \boldsymbol{x}^*_e)$ for fixed $(\mathtt{a}^e_j, m^e_j, \Delta^e_j)$, $\mathtt{a}, \in \Lambda$,
  integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $q = \Delta^e_j, j = 1, 2$;
Compute the set FP of feasible pairs $(\boldsymbol{z}, \boldsymbol{z}')$ such that $\boldsymbol{z} + \boldsymbol{z}' + \mathbf{1}_\gamma = \boldsymbol{x}^*_e$;
Compute the DAG G_1 (resp., G_2) representation of all vectors $\boldsymbol{z}$ (resp., $\boldsymbol{z}'$)
  such that $(\boldsymbol{z}, \boldsymbol{z}') \in$ FG (resp., $\boldsymbol{z}' \in$ FG);
Enumerate the set $\mathcal{P}_1$ (resp., $\mathcal{P}_2$) of paths from sources to sinks in G_1 (resp., G_2);
for each feasible pair $(\boldsymbol{z}, \boldsymbol{z}') \in$ FG **do**
  Let $P := ((\boldsymbol{z}, \boldsymbol{z}_h, \boldsymbol{y}_h, m_h), (\boldsymbol{z}_h, \boldsymbol{z}_{h-1}, \boldsymbol{y}_{h-1}, m_{h-1}), \ldots, (\boldsymbol{z}_1, \boldsymbol{z}_0, \boldsymbol{y}_0, m_0))$;
  $P' := ((\boldsymbol{z}', \boldsymbol{z}'_{h'}, \boldsymbol{y}'_{h'}, m'_{h'}), (\boldsymbol{z}'_{h'}, \boldsymbol{z}'_{h'-1}, \boldsymbol{y}'_{h'-1}, m'_{h'-1}), \ldots, (\boldsymbol{z}'_1, \boldsymbol{z}'_0, \boldsymbol{y}'_0, m'_0))$,
  Compute DAG representation G^i (resp., $(G')^i$ of each $\boldsymbol{y}_i$ (resp., $\boldsymbol{y}'_i$).
  Get a target e-component by using the trees corresponding to
    $\boldsymbol{y}_h, \boldsymbol{y}_{h-1}, \ldots, \boldsymbol{y}_0, \boldsymbol{y}'_{h'}, \ldots, \boldsymbol{y}'_0$
  Get the number of target e-components obtained by paths P and P' as
    $(n(\boldsymbol{y}_h) \times \cdots \times n(\boldsymbol{y}_0)) \times (n(\boldsymbol{y}'_{h'}) \times \cdots \times n(\boldsymbol{y}'_0))$,
    where $n(\boldsymbol{y}_i)$ (resp., $n(\boldsymbol{y}'_i)$) denote the number of graphs that can be obtained
    from $\boldsymbol{y}_i$ (resp., $\boldsymbol{y}'_i$) as explained in CompleteAlgorithmVertex
end for.

9 Canonical Representation of Fringe Trees

For a graph G, let $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. We denote by (u, v) a directed edge from vertex u to vertex v in a graph. However, we denote by uv an undirected edge between u and v in a graph, where we assume that $uv = vu$. For a vertex v, we denote by $N_G(v)$ neighbors of v. For an edge weighted graph (G, w) with weight function w and an edge $e \in E(G)$, we denote by $w(e)$ the weight of the edge e. For a rooted tree T and a non-root vertex $v \in V(T)$, we denote by $\mathrm{prt}_T(v)$ the parent of v in T. For a rooted tree T and a vertex $v \in V(T)$, we denote by $\mathrm{d}_T(v)$ depth of v in T, i.e., the length of the path between v and the root of T. When the underlaying tree T is fixed, then we simply denote parent and depth by $\mathrm{prt}(v)$ and $\mathrm{d}(v)$.

Let (T, w, λ) be a tree with n vertices, rooted at r with weight function $w : E(T) \to \{1, 2, 3\}$, a coloring function $\lambda : V(T) \to \{1, 2, \ldots, k\}$ for some k. Let $K = (T, \pi)$ be an ordered tree of T with a left-to-right ordering π on the children of each vertex, and $v_1, v_2, \ldots, v_n$ indexing on the vertices of H obtained by depth-first-search starting from the root r and visiting children following π. We define, a (color, depth)-sequence $\psi(K)$ to be the sequence

$$\psi(K) \triangleq ((\lambda(v_1), \mathrm{d}(v_1)), (\lambda(v_2), \mathrm{d}(v_2)), \ldots, (\lambda(v_n), \mathrm{d}(v_n)))),$$

a weight-sequence $\sigma(K)$ to be the sequence

$$\sigma(K) \triangleq (w_2, w_3, \ldots, w_n),$$

where $w_i = w(v_i\mathrm{prt}(v_i))$ for $i \in [2, n]$. We define a canonical representation $\mathrm{C}(T)$ of T to the pair $(\psi(K), \sigma(K))$ such that $\psi(K)$ is lexicographically maximum among the (color, depth)-sequence of all ordered trees of T and $\sigma(K)$ is lexicographically maximum among the weight- sequence of all ordered trees of T. In such a case, we call K the left-heavy representation of T. For a vertex $v \in T$, we denote by $T\langle v\rangle$ the subtree of T rooted at v that consists of v and all its descendants.

For two sequences S and S', we denote by $S \oplus S'$ the concatenation of S with S'.

We present a procedure to compute the frequencies of 2-fringes in a given set $\mathcal{G}$ of chemical graphs, where for a vertex v, we use atomic number of an atom as color $\lambda(v)$ and w as the multiplicity between the edges.

Algorithm COMPFREQ ($\mathcal{G}$)

Input: A set of chemical graphs $\mathcal{G}$.
Output: Frequencies of 2-fringe trees in $\mathcal{G}$.
Let $\mathcal{C} := \emptyset$; /* canonical representation of distinct 2-fringe trees */
for each $G \in \mathcal{G}$ **do**
  Let $G' := G$; Remove all leaves from G' in two rounds to get roots of 2-fringe trees;
  for each $v \in V(G')$ such that $N_G(v) \setminus N_{G'}(v) \neq \emptyset$ **do**
    Let (T, w, λ) be the tree rooted at v obtained by performing dfs from v to
      its descendants and T satisfies the degree condition of 2-fringe trees;
      /* $|V(T)| \leq 2d + 2$ where d is the number of children of v /*
    C[T] := CANONRECUR(T, v)
    if $C[T] \notin \mathcal{C}$ **then** $f_{C[T]} := 1$
    else $f_{C[T]} := f_{C[T]} + 1$
    endif;
    $\mathcal{C} := \mathcal{C} \cup \{C[T]\}$;
  endfor
endfor
Output $C[T]$ and $f_{C[T]}$ as the canonical representation and
  frequency of fringe tree T, respectively, for each $C[T] \in \mathcal{C}$.

Algorithm CANONRECUR(T, v)

Input: A vertex colored and edge weighted tree rooted tree (T, w, λ) and a vertex v.
Output: The canonical representation $C(T\langle v \rangle)$.
if v is a leaf **then** $C[T\langle v \rangle] := (\lambda(v), \mathrm{d}(v))$ **endif**
else
  for each child u of v **do**
    $C[T\langle u \rangle] := (\psi[T\langle u \rangle], \sigma[T\langle u \rangle]) :=$ CANONRECUR(T, u)
  endfor
  Let $v_1, v_2, \ldots v_n$ be the indexing of children of v such that for each $i \in [1, n-1]$,
    it holds that $(\psi[T\langle v_i \rangle], w(v_i v) \oplus \sigma[T\langle v_i \rangle])$ is lexicographically larger or equal to
    $(\psi[T\langle v_{i+1} \rangle], w(v_{i+1} v) \oplus \sigma[T\langle v_{i+1} \rangle])$
  Let $\psi[T\langle v \rangle] := (\lambda(v), \mathrm{d}(v)) \oplus \psi[T\langle v_1 \rangle] \oplus \cdots \oplus ((\lambda(v_n), \mathrm{d}(v_n))) \oplus \psi[T\langle v_n \rangle]$ and
    $\sigma[T\langle v \rangle] := w(v_1 v) \oplus \sigma[T\langle v_1 \rangle] \oplus \cdots \oplus w(v_n v) \oplus \sigma[T\langle v_n \rangle]$; $C[T\langle v \rangle] := (\psi[T\langle v \rangle], \sigma[T\langle v \rangle])$
endif;
Output $C[T\langle v \rangle]$ as $C(T\langle v \rangle)$.