Stable-II Partitions of Graphs

Konrad K. Dabrowski\textsuperscript{a}, Vadim V. Lozin\textsuperscript{b}, Juraj Stacho\textsuperscript{b,∗}

\textsuperscript{a}School of Engineering and Computing Sciences, Durham University, South Road, Durham DH1 3LE, United Kingdom
\textsuperscript{b}DIMAP and Mathematics Institute, University of Warwick, Coventry CV4 7AL, United Kingdom

Abstract

For a set of graphs \(\Pi\), the \textsc{Stable-\Pi} problem asks whether, given a graph \(G\), we can find an independent set \(S\) in \(G\), such that \(G - S \in \Pi\). For instance, if \(\Pi\) is the set of all bipartite graphs, \textsc{Stable-\Pi} coincides with \textsc{Vertex 3-Colourability}, and if \(\Pi\) is the set of 1-regular graphs, the problem is known as \textsc{Efficient Edge Domination}. Numerous other examples of the \textsc{Stable-\Pi} problem have been studied in the literature.

In the present contribution, we systematically study the \textsc{Stable-\Pi} problem with respect to the speed (a term meaning size) of \(\Pi\). In particular, we show that for all hereditary classes \(\Pi\) with a subfactorial speed of growth, \textsc{Stable-\Pi} is solvable in polynomial time. We then explore the problem for minimal hereditary factorial classes \(\Pi\). Contrary to the conjecture proposed in [16], the complexity of \textsc{Stable-\Pi} turns out to be polynomial for nearly all minimal hereditary factorial classes \(\Pi\). On the other hand, if we do not require \(\Pi\) to be hereditary, the complexity of the problem can jump to the NP-completeness.

Key words: stable-II partition, hereditary property, speed of graph property, factorial property, polynomial-time, NP-complete

1. Introduction

In this paper, all graphs are undirected, with no loops or parallel edges. A graph is \textit{bipartite}, \textit{co-bipartite} or \textit{split} if its vertex set can be partitioned into two independent sets, two cliques or a clique and an independent set, respectively. If \(X\) is a set of vertices in a graph \(G\), we use \(G - X\) to denote the graph obtained from \(G\) by deleting every vertex in \(X\). We write \(G[X]\) for the subgraph of \(G\) induced by \(X\), i.e. the graph \(G - (V(G) \setminus X)\).

If \(G\) and \(H\) are graphs, then \(G\) is \(H\)-free if it does not contain an induced subgraph isomorphic to \(H\). We use \(2K_2\) to denote the graph consisting of two disjoint edges, and \(P_4\) denotes the chordless path on four vertices.

Let \(\Pi\) be a graph property (or graph class), i.e. a set of graphs closed under isomorphism. A property \(\Pi\) is \textit{hereditary} if it is closed under taking induced subgraphs, and it is \textit{additive} if it is closed under taking disjoint unions of graphs.

For a property \(\Pi\), the \textsc{Stable-\Pi} problem asks, given a graph \(G\), to determine whether \(G\) has an independent set \(S\) such that \(G - S \in \Pi\). The family of \textsc{Stable-\Pi} problems has been extensively studied in the literature (see e.g. [5, 6, 7, 8, 10, 12, 13, 14, 18]) and includes many important representatives such as \textsc{Vertex 3-Colourability}, in which case \(\Pi\) is the set of all bipartite graphs, and \textsc{Efficient Edge Domination} (also known as \textsc{Dominating Induced Matching}), in which case \(\Pi\) is the set of all 1-regular graphs. Both of these examples represent algorithmically hard, i.e. NP-complete, problems. The \textsc{Stable-\Pi} problem is also NP-complete for various other properties \(\Pi\) such as forests or trivially perfect graphs [5]. More generally, the problem remains NP-complete for any additive hereditary property \(\Pi\) other than the set of edgeless graphs [15].

∗Corresponding author

Email addresses: konrad.dabrowski@durham.ac.uk (Konrad K. Dabrowski), v.lozin@warwick.ac.uk (Vadim V. Lozin), j.stacho@warwick.ac.uk (Juraj Stacho)

Preprint submitted to Elsevier

February 8, 2013
On the other hand, for some properties \( \Pi \), the \textsc{stable-II} problem can be solved in polynomial time. This is the case, for instance, if \( \Pi \) is the class of co-bipartite graphs [5] or the class of complete bipartite graphs [4]. The case of co-bipartite graphs was generalised independently in [2] and [9] to arbitrary hereditary properties \( \Pi \) which are of bounded independence number and which can be recognised in polynomial time. The case where \( \Pi \) is the class of complete bipartite graphs has also received a wide generalisation. To describe this generalisation, let us observe that the class of complete bipartite graphs is quite small. In the terminology of [3], it is \textit{subfactorial}, i.e. for any constant \( c > 0 \), \( \Pi \) has less than \( n^{cn} \) labelled graphs on \( n \) vertices, if \( n \) is sufficiently large. Subfactorial graph properties have a simple structural characterisation (see Theorem 1). This was used in [16] to prove that the \textsc{stable-II} problem is polynomial-time solvable for any subfactorial hereditary property \( \Pi \) of bipartite graphs.

In the present paper, we further generalise this result to arbitrary subfactorial hereditary properties \( \Pi \) (not necessarily of bipartite graphs). We then switch to hereditary properties with a \textit{factorial} speed of growth, i.e. those containing at least \( n^{c_1 n} \) and at most \( n^{c_2 n} \) labelled graphs on \( n \) vertices for some constants \( c_1, c_2 > 0 \), when \( n \) is sufficiently large. The family of factorial graph properties is much wider and contains many classes of theoretical or practical importance. For instance the classes of threshold graphs, line graphs, permutation graphs, and interval graphs are factorial and all classes of graphs of bounded vertex degree, of bounded clique-width and all proper minor closed graph classes have at most factorial speed of growth.

The family of factorial hereditary classes is very rich and varied, but there are only a few such classes for which the complexity of the \textsc{stable-II} problem is known. It is therefore natural to focus on the simplest classes in this family, namely those that are \textit{minimal} (when ordered by set inclusion). There are exactly nine such classes [1, 3]. Three of them are subclasses of bipartite graphs:

- \( M_1 \) \textbf{bipartite matching graphs}: graphs partitionable into two independent sets, where the edges between them form a matching (equivalently, graphs of maximum degree one)
- \( M_2 \) \textbf{bipartite almost complete graphs}: graphs partitionable into two independent sets such that each vertex has at most one non-neighbour in the opposite part
- \( M_3 \) \textbf{chain graphs}: bipartite \( 2K_2 \)-free graphs

Three other minimal factorial classes are subclasses of co-bipartite graphs: these are precisely the classes of complements of graphs in \( M_1, M_2 \) and \( M_3 \), which we denote by \( \overline{M_1}, \overline{M_2}, \) and \( \overline{M_3} \), respectively. The remaining three minimal factorial classes are subclasses of split graphs. They are also closely related to \( M_1, M_2 \) and \( M_3 \) and can be obtained from graphs in these classes by converting one of the independent sets in the bipartition into a clique. We denote these classes as follows:

- \( M_4 \) \textbf{split matching graphs}: graphs partitionable into a clique and an independent set, where the edges between them form a matching
- \( \overline{M_4} \) \textbf{complements of split matching graphs}: graphs partitionable into a clique and an independent set so that each vertex has at most one non-neighbour in the opposite part
- \( M_5 \) \textbf{threshold graphs}: split \( P_4 \)-free graphs

It is known that \textsc{stable-}\( M_1 \) is an NP-complete problem [17], while \textsc{stable-}\( M_2 \) is solvable in polynomial time [5]. For the remaining seven minimal factorial classes, the complexity of the problem was unknown and we study it in the present paper.

The borderline between factorial and subfactorial properties was also studied in [19] for the following problem associated with a hereditary class \( \Pi \) of bipartite graphs: given a bipartite graph \( G \), find the largest induced subgraph of \( G \) that belongs to \( \Pi \). Yannakakis [19] showed that this problem is solvable in polynomial time if \( \Pi \) is a subfactorial hereditary class, and is NP-hard otherwise (except for the case when \( \Pi \) coincides with the class of all bipartite graphs, in which case the problem is trivial). Inspired by this result, Lozin conjectured [16] that the \textsc{stable-II} problem is NP-complete for all hereditary factorial classes of bipartite graphs, including the three minimal hereditary factorial classes. Contrary to this conjecture, we
show that STABLE-Π is solvable in polynomial time for nearly all minimal hereditary factorial classes Π (not necessarily bipartite).

Let us emphasise that these nine minimal classes of graphs are hereditary and most of the instances of the STABLE-Π problem that have been studied in the literature deal with hereditary properties Π. On the other hand, some important examples of the problem appear in the context of non-hereditary properties Π. We already mentioned EFFICIENT EDGE DOMINATION, which is equivalent to STABLE-Π when Π is the set of 1-regular graphs. We denote the class of 1-regular graphs by $M_1$. Observe that this set is a restriction of the class $M_1$. More precisely, $M_1$ is the hereditary closure of the set of 1-regular graphs (i.e. it is the set containing all 1-regular graphs and all their induced subgraphs). In the same spirit, we define $M_2$ to be the class of graphs partitionable into two independent sets such that each vertex has exactly one non-neighbour in the opposite part and define $M_4$ to be the class of graphs partitionable into a clique and an independent set such that every vertex in one part has exactly one neighbour in the opposite part. As before, we write $M_1^S$, $M_2^S$ and $M_4^S$ to denote the classes of graphs whose complements are in $M_1$, $M_2$ and $M_4$, respectively.

We find that for some minimal factorial classes Π for which STABLE-Π can be solved in polynomial time, the restriction to Π$^S$ leads to an NP-complete problem. A summary of our results is given in Table 1.

### 2. Preliminaries

A graph property, or graph class, is any set Π of simple graphs closed under isomorphism. The graph-complement Π$^c$ of a property Π is defined as $Π^c = \{ G^c \mid G \in Π \}$. A graph property is hereditary if it is closed under vertex removal, or equivalently, under taking induced subgraphs. A hereditary graph property Π is factorial if there exist constants $c_1, c_2, N$ such that $n^{c_1 n} \leq |Π_n| \leq n^{c_2 n}$ when $n > N$, where $Π_n$ denotes the set of n-vertex labelled graphs in Π. A class is subfactorial if for every $c > 0$, $|Π_n| \leq n^{cn}$ when $n$ is sufficiently large.

The structure of subfactorial classes is rather simple and can be characterised as follows.

**Theorem 1.** [1, 3] For every subfactorial hereditary class Π, there exists a constant $k$ (depending only on Π) such that for every graph $G \in Π$, there exists a partition of $V(G)$ into at most $k$ subsets $V_1, \ldots, V_k$, where each subset $V_i$ is either an independent set or a clique in $G$, and for any two distinct subsets $V_i, V_j$, there are either no edges or all possible edges between the vertices in $V_i$ and the vertices in $V_j$.

### 3. Subfactorial properties

**Theorem 2.** For any subfactorial hereditary property Π, the STABLE-Π problem is solvable in polynomial time.

<table>
<thead>
<tr>
<th>Π</th>
<th>STABLE-Π</th>
<th>STABLE-Π$^S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_1$</td>
<td>NP-C</td>
<td>Thm 17</td>
</tr>
<tr>
<td>$M_1$</td>
<td>P</td>
<td>Thm 6</td>
</tr>
<tr>
<td>$M_2$</td>
<td>P</td>
<td>Thm 14</td>
</tr>
<tr>
<td>$M_2$</td>
<td>P</td>
<td>Thm 6</td>
</tr>
<tr>
<td>$M_3$</td>
<td>open</td>
<td>n/a</td>
</tr>
<tr>
<td>$M_3$</td>
<td>P</td>
<td>Thm 6</td>
</tr>
<tr>
<td>$M_4$</td>
<td>P</td>
<td>Thm 7</td>
</tr>
<tr>
<td>$M_4$</td>
<td>P</td>
<td>Thm 8</td>
</tr>
<tr>
<td>$M_5 = M_5$</td>
<td>P</td>
<td>[5]</td>
</tr>
</tbody>
</table>

Table 1: Summary of complexity results.
Theorem 4. Let $k$ be the constant associated with the class $\Pi$. We call any partition of $R$ satisfying Theorem 1 canonical and call the subsets in a canonical partition bags.

We start by picking a representative for each bag. There are $O(n^k)$ ways to do so. Once the set of representatives is fixed, which is our current set $R$, the adjacencies between the bags are defined by the adjacencies between their representatives. For each choice of at most $k$ representatives, there are at most $2^k$ ways to choose the type for each bag (a clique or an independent set). Without loss of generality we may assume that for each vertex $v \in V - R$ there is at most one candidate bag for the inclusion of $v$, since otherwise any two “similar” bags can be replaced by a single bag containing both of them. If there is no candidate bag for $v$, we move it to $S$.

For the vertices $v$ not in $R \cup S$ we proceed as follows: If $v$ has a conflict in $S$ (i.e. has a neighbour in $S$) we move it to the respective bag of $R$, and if $v$ has a conflict in $R$ (i.e. moving it to its candidate bag in $R$ makes the partition of $R$ non-canonical) we move it to $S$. If no vertex outside of $R \cup S$ has a conflict in $S$ or $R$, then the rest of the task can be solved by a reduction to the 2SAT problem.

To this end, we associate with each vertex $v \notin R \cup S$ a Boolean variable $x_v$. For any two vertices $u, v \notin S \cup R$, we create a set of clauses in the following way. If $u$ and $v$ cannot both appear in $R$ (because, for instance, they are adjacent, but their candidate bags are not) we create the clause $x_u \lor x_v$, and if they cannot both appear in $S$ we create the clause $\neg x_u \lor \neg x_v$. It is easy to verify that the set of clauses created in this way is satisfiable if and only if there is a proper partition of $G$ in which every vertex $v$ with $x_v = \text{true}$ is placed in $S$ and the remaining vertices are placed in $R$. □

4. Minimal factorial properties

In this section, we discuss the complexity of Stable-$\Pi$ for minimal factorial hereditary classes $\Pi$. We investigate each case as set out in the introduction.

The following cases have already been established in the literature.

Theorem 3. [17] The Stable-$\mathcal{M}_1$ problem is NP-complete.


Further results in this section are based on the notion of Sparse-Dense partitions.

Theorem 5. (Sparse-Dense Theorem) [2, 9] For all positive integers $k, l$, there exists a polynomial time algorithm that, given a graph $G$, constructs all partitions of its vertex set into sets $A, B$ such that $G[A]$ contains no independent set of size $k$ and $G[B]$ contains no clique of size $l$.

Namely, there are at most $n^{2R(k,l)-2}$ such partitions of an $n$-vertex graph $G$ and all can be enumerated in time $O(n^{2R(k,l) + \max(k,l)})$, where $R(k,l)$ denotes the Ramsey number of $k$ and $l$.

Theorem 6. The Stable-$\mathcal{M}_1$, Stable-$\mathcal{M}_2$, and Stable-$\mathcal{M}_3$ problems are solvable in polynomial time.

Proof. Let $\Pi \in \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\}$. All three problems ask to partition vertices of the input graph $G$ into one independent set $V_1$, and a co-bipartite graph $V'_1$ (consisting of two cliques $V_2$ and $V_3$). By Theorem 5, there are only polynomially many such partitions of $V(G)$ and all of them can be found in polynomial time. For each such partition, we test whether the co-bipartite subgraph of $G$ induced by $V'_1$ is in $\Pi$. This yields a polynomial-time algorithm. □

The following theorems are proved in a similar way to how Theorem 4 was proved in [5].

Theorem 7. The Stable-$\mathcal{M}_4$ problem is solvable in polynomial time.
Proof. We rephrase the problem as: given a graph $G$, decide whether the vertices of $G$ can be partitioned into three sets $V_1, V_2, V_3$ such that $V_3$ is a clique, $V_1$ and $V_2$ are independent sets and every vertex in $V_2$ has at most one neighbour in $V_3$ and vice-versa.

Let $G$ be the input graph. By Theorem 5, we can find, in polynomial time, the collection $P$ of all partitions of the vertex set of $G$ into a clique $C$ and a set $X$ such that $G[X]$ contains no clique of size three. Note that if $G$ admits a \textsc{stable-M4} partition $V_1, V_2, V_3$, then the partition $C = V_3, X = V_1 \cup V_2$ is a partition in $P$. Thus to solve the problem, we try all partitions $C, X$ in $P$ by setting $V_3 = C$ and testing whether $X$ can be split into $V_1, V_2$ so that $V_1, V_2, V_3$ is a \textsc{stable-M4} partition of $G$.

Let $C, X$ be a partition from $P$. We construct the following instance $I$ of $\textsc{2Sat}$.

1. Create a variable $x_v$ for every vertex $v \in X$,
2. for every edge $uv \in E(G[X])$, add the clauses $(x_u \lor x_v)$ and $(\overline{x_u} \lor \overline{x_v})$,
3. for every pair of vertices $u, v \in X$ with a common neighbour in $C$, add the clause $(x_u \lor x_v)$, and
4. for every vertex $v \in X$ such that $v$ has at least two neighbours in $C$, add the clauses $(\overline{x_v} \lor a)$ and $(\overline{x_v} \lor \overline{a})$, where $a$ is a new variable.

We claim that $I$ has a satisfying assignment if and only if $G$ admits a \textsc{stable-M4} partition $V_1, V_2, V_3$ such that $V_3 = C$ and $V_1 \cup V_2 = X$.

Suppose that the instance $I$ has a satisfying truth assignment $\varphi$. Namely, $\varphi$ is a mapping from the variables of $I$ to $\{true, false\}$ such that in every clause $C_i$, there is at least one literal that $\varphi$ evaluates to $true$ (where the value $\varphi(\overline{x})$ is defined as the negation of $\varphi(x)$, for any variable $x$).

Define $V_1 = \{v \mid \varphi(x_v) = false\}$ and $V_2 = \{v \mid \varphi(x_v) = true\}$. We claim that $V_1, V_2, V_3$ is a \textsc{stable-M4} partition of $G$. Indeed, by (ii), $V_1$ and $V_2$ are independent sets; by (iii), no two vertices in $V_2$ have a common neighbour in $V_3$; and by (iv), every vertex from $V_2$ has at most one neighbour in $V_3$.

Conversely, let $V_1, V_2, V_3$ be a \textsc{stable-M4} partition of $G$ where $V_3 = C$. We define a truth assignment for $I$ as follows. We set $\varphi(x_v) = false$ if $v \in V_1$ and $\varphi(x_v) = true$ if $v \in V_2$. For each of the new variables $a$ defined in (iv) above, we set $\varphi(a) = true$. We claim that $\varphi$ is a satisfying truth assignment for $I$. Indeed, all clauses defined in (ii) are satisfied, since $V_1$ and $V_2$ are independent sets. Also, all clauses defined in (iii) are satisfied since every vertex in $V_3$ has at most one neighbour in $V_2$. Similarly, every vertex in $V_2$ has at most one neighbour in $V_3$ implying that all clauses in (iv) are satisfied. Thus $I$ is satisfied by $\varphi$. This concludes the proof.

A similar argument works for the complementary class and results in the following theorem.

**Theorem 8.** The \textsc{stable-M4} problem is solvable in polynomial time.

Proof. Similarly to the proof of Theorem 7, we can rephrase the problem as: given a graph $G$, decide whether the vertices of $G$ can be partitioned into three sets $V_1, V_2, V_3$ such that $V_3$ is a clique, $V_1$ and $V_2$ are independent sets and every vertex in $V_2$ has at most one non-neighbour in $V_3$ and vice-versa.

Again, defining $P$ as before, we solve the problem by trying all partitions $C, X$ in $P$. For each such partition we set $V_3 = C$ and test whether $X$ can be split into $V_1, V_2$ so that $V_1, V_2, V_3$ is a \textsc{stable-M4} partition of $G$.

Let $G'_C$ be the graph obtained from $G$ by complementing (i.e. replacing edges by non-edges and vice versa) the edges between $C$ and $X$. Now $G$ has a \textsc{stable-M4} partition with $V_3 = C$ if and only if $G'_C$ has a \textsc{stable-M4} partition with $V_3 = C$. Indeed, if $V_1 \cup V_2$ is a partition of $X$, then $G'[C]$ is a clique and $G'[V_1], G'[V_2]$ are independent sets if and only if $G'_C[C]$ is a clique and $G'_C[V_1], G'_C[V_2]$ are independent sets. Further, each vertex in $V_2$ (resp. $V_3$) has at most one non-neighbour in $V_3$ (resp. $V_2$) in $G$ if and only if it has at most one neighbour in $V_3$ (resp. $V_2$) in $G'_C$.

We now reduce the problem to an equivalent instance of $\textsc{2Sat}$ as in the proof of Theorem 7. This concludes the proof.

We are left with the case of the \textsc{stable-M2}, which needs more work. We solve this in the following section.
Algorithm 1: Reduction algorithm

**Input:** Instance \((G, \ell)\) where \(G\) is a graph and \(\ell(v) : V(G) \rightarrow 2^{\{1,2,3\}}\)

**Output:** A reduced instance \((G, \ell)\)

1. for \(\alpha \in \{1,2,3\}\) do
   2. if for \(u \in U^\ell_{\{\alpha\}}\), there exists \(v \in V(G) \setminus N(u)\) with \(\alpha \in \ell(v)\) then
      remove \(\alpha\) from \(\ell(v)\) and goto 1
   3. for \((\alpha, \beta) \in \{(2,3), (3,2)\}\) do
      4. if for \(u \in U^\ell_{\{\alpha\}}, \) there exists \(v \in N(u) \cap U^\ell_{\{\beta\}}\) then
         for all \(w \in N(u) \setminus \{v\}\) with \(\beta \in \ell(w)\), remove \(\beta\) from \(\ell(w)\)
         for all \(w \in N(v) \setminus \{u\}\) with \(\alpha \in \ell(w)\), remove \(\alpha\) from \(\ell(w)\)
         remove \(u, v\) from \(G\) and goto 1
      5. if there exists \(v \in U^\ell_{\{1,\beta\}}\) with \(|N(v) \cap U^\ell_{\{\alpha\}}| \geq 2\) then
         remove \(\beta\) from \(\ell(v)\) and goto 1
      6. if for \(u \in U^\ell_{\{\alpha\}}, \) there are \(v, w \in N(u) \cap U^\ell_{\{1,\beta\}}\) where\n         \((N(v) \setminus N(w)) \cap U^\ell_{\{1,\alpha\}} \neq \emptyset\) then
         remove \(\beta\) from \(\ell(v)\) and goto 1
      7. if for \(u \in U^\ell_{\{\alpha\}}, \) there are \(v, w \in N(u) \cap U^\ell_{\{1,\beta\}}\) and \(x \in U^\ell_{\{1,\alpha\}}\) with \(v, w \notin N(x)\) then
         remove \(1\) from \(\ell(x)\) and goto 1
      8. if for \(u \in V(G)\) with \(1 \in \ell(u)\), the set \(U^\ell_{\{1,\alpha\}} \setminus N(u)\) is not a clique then
         remove \(1\) from \(\ell(u)\) and goto 1
      9. if for \(u \in V(G)\) with \(\beta \in \ell(u)\), the subgraph \(G[N(u) \cap U^\ell_{\{1,\alpha\}}]\) contains\n         an induced 4-cycle, \(2K_2\), or \(P_4\) then
         remove \(\beta\) from \(\ell(u)\) and goto 1
   10. return \((G, \ell)\)

4.1. The Stable-\(\mathcal{M}_2\) problem

In this section, we prove that the Stable-\(\mathcal{M}_2\) problem is solvable in polynomial time. We cast the problem for the complement and solve (in polynomial time) a more general version with lists as follows.

An instance of the problem is a pair \((G, \ell)\) where \(G\) is a graph and \(\ell : V(G) \rightarrow 2^{\{1,2,3\}}\). We say that \(\ell(v)\) is the list belonging to the vertex \(v\). For \(S \subseteq \{1,2,3\}\), we let \(U^\ell_S\) denote the set of vertices in \(G\) with \(\ell(v) = S\).

Given an instance \((G, \ell)\), we seek to partition \(V(G)\) into three cliques \(V_1, V_2, V_3\) such that

- each vertex in \(V_2\) has at most one neighbour in \(V_3\),
- each vertex in \(V_3\) has at most one neighbour in \(V_2\), and
- for all \(\alpha \in \{1,2,3\}\), each \(v \in V_\alpha\) satisfies \(\alpha \in \ell(v)\).

If such a partition exists, we call it a solution for \((G, \ell)\). Note that if the list of some vertex is empty, then there is no solution for the problem instance. Thus for the rest of the proof, we assume that \(U^\ell_{\emptyset} = \emptyset\).

To solve the problem, we consider several special cases and reduce the general case to these cases in polynomial time.

First, we consider the procedure depicted in Algorithm 1. We say that an instance \((G, \ell)\) is reduced, if it is the result of Algorithm 1.

We have the following claim.

**Lemma 9.** Let \((G, \ell)\) be an instance and let \((G', \ell')\) be the result of applying Algorithm 1 to \((G, \ell)\). Then there exists a solution for \((G, \ell)\) if and only if there exists a solution for \((G', \ell')\).
PROOF. Note that if $x \in U_i^\ell$ for some $i \in \{1, 2, 3\}$, then in any solution $(V_1, V_2, V_3)$ of the instance, we have $x \in V_i$. Using this we justify the reductions rules as follows.

**Line 2**: Let $\alpha \in \{1, 2, 3\}$. Since $V_\alpha$ must be a clique in any solution, if $u \in U_\alpha^\ell$ and $u, v$ are not adjacent, then $v \notin V_\alpha$ for any solution for $(G, \ell)$.

In the remainder of the proof, we have $\alpha = 2$ and $\beta = 3$, or $\alpha = 3$ and $\beta = 2$.

**Line 4**: If $u, v$ are adjacent for some $u \in U_\alpha^\ell$ and $v \in U_\beta^\ell$, then in any valid solution, these must be two matched vertices of $V_2$ and $V_3$. In this case $v$ must be the unique neighbour of $u$ in $V_\beta$ and $u$ must be the unique neighbour of $v$ in $V_\alpha$. We can therefore remove either $\alpha$ or $\beta$ from the list of each vertex in $N(u) \cup N(v) \setminus \{u, v\}$, as appropriate. We then remove $u$ and $v$ from $G$. The resulting instance has a solution if and only if the original one does.

**Line 5**: In any solution, if $v \in V_\beta$, then $v$ can have at most one neighbour in $V_\alpha$.

**Line 6**: Suppose $u \in U_\alpha^\ell$, such that $v, w \in N(u) \cap U_\beta^\ell$ and $z \in (N(v) \setminus N(w)) \cap U_\alpha^\ell$. If there were a solution in which $v \in V_\beta$, then since $u \in V_\alpha$ and every vertex in $V_\alpha$ can have at most one neighbour in $V_\beta$ and vice versa, we must have $w, z \in V_1$. But this is impossible, since $w, z$ are not adjacent. This contradiction implies that $v$ cannot be in $V_\beta$.

**Line 7**: Suppose $u \in U_\alpha^\ell, x \in U_\alpha^\ell$ and $v, w \in (N(u) \setminus N(x)) \cap U_\beta^\ell$. Then in any solution we must have $u \notin V_\alpha$. Since $u$ has at most one neighbour in $V_\beta$, at least one of $v, w$ must be in $V_1$. But $V_1$ is a clique and $v, w$ are non-adjacent to $x$. Thus $x \notin V_1$.

**Line 8**: Suppose $u \in V(G)$ with $1 \in \ell(u)$ and $v, w \in U_\alpha^\ell \setminus N(u)$ with $v, w$ non-adjacent. Since for any solution, $V_1$ must be a clique for $i \in \{1, 2, 3\}$, exactly one of $v, w$ must be in $V_1$ and the other in $V_\alpha$. But $u$ is non-adjacent to both $v$ and $w$, so $u \notin V_1$.

**Line 9**: Suppose $\beta \in \ell(u)$. In any solution, if $u \in V_\beta$ then $N(u) \cap V_1$ must be a clique and $u$ can have at most one neighbour in $V_\alpha$. The 4-cycle, $2K_2$ and $P_4$ are neither cliques, nor are they partitionable into a clique and a single vertex. Thus if any of these three graphs is an induced subgraph of $N(u) \cap U_\alpha^\ell$, then any solution must satisfy $u \notin V_\beta$.

Note that Algorithm 1 has polynomial running time. This allows us to assume that the instance we consider is always reduced. (If not, we use Algorithm 1 to produce an equivalent reduced instance.)

Assuming this, we consider the some special cases of the problem, which we will later use as steps in finding a solution for the general problem.

**Lemma 10.** If there exists a solution $(V_1, V_2, V_3)$ for the reduced instance $(G, \ell)$, such that there is no edge between a vertex in $V_2$ and a vertex in $V_3$, it can be found in polynomial time.

**Proof.** This amounts to finding a partition of $\overline{G}$ into an independent set and a complete bipartite graph, in a way that respects the lists of the vertices. This can been solved in polynomial time [9].

**Lemma 11.** If $U_1^\ell(2,3) = U_2^\ell(2,3) = \emptyset$, and $U_1^\ell(1,2) = \emptyset$ or $U_2^\ell(1,3) = \emptyset$, and the instance is reduced, the problem can be solved in polynomial time.

**Proof.** We may assume by symmetry that $U_1^\ell(1,3) = \emptyset$ and we reduce the problem to an instance of $2\text{SAT}$ constructed as follows.

• For each vertex $x \in U_i^\ell(1,2)$, introduce a new variable $v_x$.
• For all $z \in U_1^\ell(1,3)$ and all $x, y \in N(z) \cap U_1^\ell(1,2)$, add the clause $(\neg v_x \lor \neg v_y)$.
• For all $x, y \in U_1^\ell(1,2)$ with $xy \notin E(G)$, add the clauses $(v_x \lor v_y), (\neg v_x \lor \neg v_y)$.

Since $(G, \ell)$ is a reduced instance, it has a solution if and only if the above instance of $2\text{SAT}$ is satisfiable. In particular, if $\phi$ is a satisfying assignment, the following sets $(V_1, V_2, V_3)$ form a solution for $(G, \ell)$.

$V_1 = U_1^\ell(1) \cup \{x \mid \phi(v_x) = \text{false}\}$  $V_2 = U_2^\ell(1,2) \cup (U_2^\ell(1,2) \setminus V_1)$  $V_3 = U_3^\ell(3)$
Lemma 12. If $U^\ell_{(1,2,3)} = U^\ell_{(2,3)} = \emptyset$ and $U^\ell_{(1,2)}$, $U^\ell_{(1,3)}$ are cliques of $G$, and the instance is reduced, the problem can be solved in polynomial time.

Proof. We show that the following is a solution for $(G, \ell)$.

$$V_1 = U^\ell_{(1)} \cup U^\ell_{(2,1)} \cup \bigcup_{u \in U^\ell_{(2)}} (N(u) \cap U^\ell_{(1,3)}), \quad V_2 = U^\ell_{(2)} \quad V_3 = U^\ell_{(3)} \cup (U^\ell_{(1,3)} \setminus V_1)$$

Indeed, note that the instance $(G, \ell)$ is reduced. By Line 2 of Algorithm 1 and the fact that $U^\ell_{(1,3)}$ is a clique, we conclude that $V_2$ and $V_3$ must be cliques. By Line 4 of Algorithm 1 and the definition of $V_1$ and $V_3$, every vertex in $V_2$ has at most one neighbour in $V_3$. By Lines 4 and 5 of Algorithm 1, each vertex of $V_3$ has at most one neighbour in $V_2$. By Line 2 of Algorithm 1 and since $U^\ell_{(1,2)}, U^\ell_{(1,3)}$ are cliques, we need only verify that every vertex in $V_1 \cap U^\ell_{(1,2)}$ is adjacent to every vertex in $V_1 \cap U^\ell_{(1,3)}$. We therefore assume that these sets are not empty. Let $u \in U^\ell_{(2)}$ and $v, w \in N(u) \cap U^\ell_{(1,3)}$. By Line 7 of Algorithm 1, any vertex in $U^\ell_{(1,2)}$ must be adjacent to at least one of $v$ or $w$. But by Line 6 of Algorithm 1, the vertices $v, w$ have the same neighbourhood in $U^\ell_{(1,2)}$. Thus every vertex of $U^\ell_{(1,2)}$ must be adjacent to every vertex of $V_1 \cap U^\ell_{(1,3)}$. We therefore conclude that $V_1$ is indeed a clique. $\square$

We can now generalise Lemmas 11 and 12 as follows.

Lemma 13. If $U^\ell_{(1,2,3)} = U^\ell_{(2,3)} = \emptyset$, and the problem instance is reduced, the problem can be solved in polynomial time.

Proof. Assume that $U^\ell_{(1,2,3)} = U^\ell_{(2,3)} = \emptyset$, but Lemma 11 does not apply. Thus $U^\ell_{(1,2)} \neq \emptyset$ and $U^\ell_{(1,3)} \neq \emptyset$.

We fix any $u \in U^\ell_{(1,2)}$. Then we either do nothing, or choose $w \in N(u) \cap U^\ell_{(1,3)}$ and set $\ell(w) = \{3\}$. After that, we remove 3 from $\ell(v)$ for each $v \in N(u)$ that belongs to a non-trivial ($\geq 2$ vertices) connected component of $G[U^\ell_{(1,3)}]$ unless that component contains $w$ (if $w$ exists). We then apply Algorithm 1 to ensure that we have a reduced instance.

If after these modifications $U^\ell_{(1,3)}$ is still non-empty, we similarly fix $w' \in U^\ell_{(1,3)}$, do nothing or set $\ell(w') = \{2\}$ for some $w' \in N(w') \cap U^\ell_{(1,2)}$, and then remove 2 from $\ell(v)$ for each $v \in N(w') \cap U^\ell_{(1,2)}$ in a non-trivial component of $G[U^\ell_{(1,2)}]$ unless that component contains $w'$ (if $w'$ exists). Afterwards, we again apply Algorithm 1 to ensure that we have a reduced instance.

We try all possible choices for $w$ and $w'$, creating $O(n^2)$ instances. It follows that the initial instance has a solution if and only if one of these $O(n^2)$ instances has.

Consider the $O(n^2)$ instances produced in this way from the initial instance $(G, \ell)$. First, we show that $(G, \ell)$ has a solution if and only if (at least) one of the $O(n^2)$ instances has a solution.

Clearly, if one of the $O(n^2)$ instances has a solution, then this is also a solution for $(G, \ell)$, since during the construction of the instances, we only remove elements from lists.

Conversely, let $V_1, V_2, V_3$ be a solution for $(G, \ell)$. Let $H = G[U^\ell_{(1,3)}]$, i.e. $H$ denotes the subgraph of $G$ induced by $U^\ell_{(1,3)}$, and consider the vertex $u \in U^\ell_{(1,2)}$ that we fix.

Case(i): Suppose that $u \in V_1$. There are two possibilities to consider. First, suppose that there exists a neighbour of $u$ that is in $V_3$ and also in some non-trivial connected component of $H$. Consider the instance where we choose $w$ to be this neighbour. (We shall henceforth refer to it as the "modified" instance.) In this instance, we remove 3 from each neighbour of $u$ in $V(H) = U^\ell_{(1,3)}$ that belongs to a non-trivial connected component of $H$ unless that component contains $w$.

We claim that each such neighbour $v$ belongs to $V_1$. Suppose otherwise. Then $v$ belongs to $V_3$, since $\ell(v) = \{1, 3\}$. Recall that $v$ is in a non-trivial connected component of $H$. Thus it has a neighbour $z$ in $H$. We conclude that $z$ is non-adjacent to $v$ in $H$, and hence, in $G$. If $z$ is also non-adjacent to $u$, then $z$ can
be neither in $V_1$ nor in $V_3$, as these are both cliques. But then $V_1, V_2, V_3$ cannot be a solution for $(G, \ell)$ as $\ell(z) = \{1,3\}$. So, we conclude that $z$ is adjacent to $u$.

Now, recall that $w$ is also in a non-trivial connected component of $\overline{H}$. So, $w$ has a neighbour $x$ in this component, and we conclude that $xw \notin E(G)$. This implies $ux \in E(G)$ as otherwise $V_1, V_2, V_3$ is not a solution. But now $x, z, w, v$ induce a $4$-cycle in the neighbourhood of $u$, which is impossible by Line 9 of Algorithm 1. (For this, recall that $(G, \ell)$ is a reduced instance and that the connected component of $\overline{H}$ containing $w$ and $x$ is different from the one containing $v$ and $z$.)

This proves that $V_1, V_2, V_3$ is also a solution to the modified instance. As this is one of the $O(n^2)$ instances, we are done.

So, we may assume that each neighbour of $u$ in $V_3 \cap V(H)$ is itself a connected component (isolated vertex) of $\overline{H}$. In this case, we consider the instance where we do not choose $w$ (referred to as the “modified” instance). In this instance, we remove $3$ from each neighbour of $u$ in $V(H)$ that belongs to a non-trivial connected component of $\overline{H}$. By our assumption, this does not modify the lists of those neighbours of $u$ that are in $V_3 \cap V(H)$. Thus $V_1, V_2, V_3$ is a solution to the modified instance, and we are done.

**Case (ii):** Suppose that $u \in V_2$. If $u$ has a neighbour in $V_3 \cap V(H)$, consider the instance where $w$ is chosen to be this neighbour (referred to as the “modified” instance). In this instance, we remove $3$ from each neighbour of $u$ in a non-trivial connected component of $\overline{H}$ unless that component contains $w$. Clearly, any such vertex $v$ cannot belong to $V_3$, since then $u$ would have two neighbours in $V_3$, which is impossible. Thus $V_1, V_2, V_3$ is also a solution to the modified instance, and we are done.

Finally, suppose that $u$ has no neighbour in $V_3 \cap V(H)$, and consider the instance where we do not choose $w$. Again, we remove $3$ from every neighbour of $u$ in a non-trivial component of $\overline{H}$, and conclude that $V_1, V_2, V_3$ is a solution to this modified instance, since we assume that $N(u) \cap V(H) \cap V_3 = \emptyset$. This completes all cases.

This proves that one of the choices for $w$ must succeed if $(G, \ell)$ has a solution. By a symmetric argument, it follows that, for an appropriate choice of $w$, one of the choices for $w'$ (if at all we consider $w'$) must also succeed. This concludes the first argument.

For the second argument, consider one of the $O(n^2)$ instances $(G^+, \ell^+)$. We constructed this instance from the initial instance $(G, \ell)$, by fixing a vertex $u$ and choosing $w$ (or not), and then fixing a vertex $u'$ (if possible) and choosing $w'$ (or not). We also reduced this instance using Algorithm 1.

We now prove that $U^+_{\{1,2\}}$ and $U^+_{\{1,3\}}$ are both cliques of $G$, i.e. that Lemma 12 can be applied. Suppose otherwise, and assume first that $U^+_{\{1,3\}}$ contains non-adjacent vertices $v, v'$. As $\ell^+$ is a reduction of $\ell$ and since $U_{\{1,2\}}^+ = \emptyset$, we conclude that $v, v'$ are also vertices in $U_{\{1,3\}}^+$. Again, let $H$ denote the graph $G[U_{\{1,3\}}^+]$.

First, we observe that $u$ is adjacent to at least one of $v, v'$. Indeed, if $u$ is non-adjacent to both $v$ and $v'$, then $1$ was removed from $\ell(u)$ in Line 8 of Algorithm 1 (recall that $(G, \ell)$ is a reduced instance). This is impossible as $\ell(u) = \{1, 2\}$. By symmetry, we shall assume that $u$ is adjacent to $v$.

Now, if $w$ was not chosen when constructing $(G^+, \ell^+)$, then $3$ was removed from all neighbours of $u$ in non-trivial connected components of $\overline{H}$. One of these components contains both $v$ and $v'$ as they are non-adjacent, and so $3$ was removed from $\ell(v)$ when constructing $\ell^+$ (recall that we assume that $u$ is adjacent to $v$). However, this is impossible, since $\ell^+(v) = \{1, 3\}$. We similarly arrive at a contradiction when $w$ is chosen, but it is not a vertex of the connected component of $\overline{H}$ containing $v$. So we conclude that $w$ was chosen from the connected component of $\overline{H}$ containing $v$. But now, we have that either $v = w$, or, since $(G^+, \ell^+)$ is reduced, $1$ or $3$ was removed from $\ell(v)$ in Line 2 at some point when running Algorithm 1 to produce the instance $(G^+, \ell^+)$. This is, of course, impossible as $\ell(w) = \{3\}$ and $\ell^+(v) = \{1, 3\}$. This concludes the argument for $U^+_{\{1,3\}}$.

The argument for $U^+_{\{1,2\}}$ is similar, using $u'$ and $w'$. Finally, note that if $u'$ (and hence $w'$) cannot be chosen because the first modification of lists removed all candidates, then the Lemma 11 can be applied.

We are ready to discuss the general case and prove the main theorem of this section.

**Theorem 14.** The **Stable-$M_2$ problem** is solvable in polynomial time.
Proof. First, we test whether or not we are in the situation of Lemma 10. If so, we find a solution for \((G, \ell)\) using \([9]\). If not, we conclude that if there is a solution \((V_1, V_2, V_3)\) for \((G, \ell)\), then there must exist \(u \in V_2\) and \(v \in V_3\) with \(uv \in E(G)\). We try all possible choices for such a pair \(u, v\). This reduces the problem to solving \(O(n^2)\) separate instances. For each such choice \(u, v\), we set \(\ell(u) = \{2\}, \ell(v) = \{3\}\), and run Algorithm 1. If the list of some vertex becomes empty, we reject this choice of \(u, v\). Otherwise, we observe that the resulting reduced instance \((G', \ell')\) satisfies \(U_{(1, 2, 3)}'' = U''_{(2, 3)} = \emptyset\). So we can apply Lemma 13 to \((G', \ell')\), which determines in polynomial time if there is a solution for \((G, \ell)\). This concludes the proof. \(\Box\)

5. Restricted Minimal Factorial Properties

First, we briefly examine the polynomial-time cases. Using essentially the same arguments as in the proof of Theorem 6, we obtain the following theorem.

Theorem 15. The \(\text{STABLE-M}_1^3\) and \(\text{STABLE-M}_2^3\) problems are solvable in polynomial time.

All the remaining cases are hard. We discuss them in separate claims. All the subsequent proofs will be essentially along the same lines and based on the following useful lemma.

Lemma 16. Any non-empty instance of \(\text{ONE-IN-THREE-3SAT}\) can be transformed in polynomial time to an equivalent instance of \(\text{ONE-IN-THREE-3SAT}\) such that

(i) There is no clause of the form \((X \lor X \lor Y)\) or \((X \lor \overline{X} \lor Y)\) where \(X, Y\) are (not necessarily distinct) literals.

(ii) If \(X\) appears in some clause, then \(\overline{X}\) also appears in some clause.

(iii) Every literal appears at least twice in the instance.

(iv) There are at least 4 clauses and at least 4 variables in the instance.

Proof. Apply the following steps in order. First, for each clause of the form \((X \lor X \lor Y)\), replace it by the clauses \((u \lor v \lor X), (\overline{v} \lor \overline{w} \lor X), (w \lor z \lor Y), (\overline{w} \lor \overline{z} \lor Y)\), where \(u, v, w, z\) are new variables. Next, for each clause of the form \((X \lor \overline{X} \lor Y)\), replace it by the clauses \((u \lor v \lor Y), (\overline{v} \lor \overline{w} \lor Y), (v \lor \overline{v} \lor z), (v \lor \overline{z} \lor w)\), where \(u, v, w, z\) are new variables. Then, for each literal \(X\), add the clauses \((u \lor v \lor X), (\overline{v} \lor \overline{w} \lor X), (\overline{w} \lor \overline{z} \lor X), (v \lor \overline{v} \lor z), (v \lor \overline{z} \lor w)\), where \(u, v, w, z\) are new variables. Note that since the original instance was non-empty, the new instance must now have at least 4 clauses and at least 4 variables. Finally, make a copy of each clause, i.e. make each clause appear twice in the instance.

It is easy to see that the instance produced in this way is equivalent to the original instance and satisfies all the conditions of the lemma. \(\Box\)

Theorem 17. The \(\text{STABLE-M}_4^3\) problem is \(\text{NP-complete}\).

Proof. We can rephrase the problem as follows: given a graph \(G\), decide whether the vertices of \(G\) can be partitioned into 3 sets \(V_1, V_2, V_3\) such that \(V_3\) is a clique, \(V_1\) and \(V_2\) are independent sets and the edges between \(V_2\) and \(V_3\) form a perfect matching.

The proof proceeds by reduction from \(\text{ONE-IN-THREE-3SAT}\). Consider an instance \(I\) of the problem, namely the instance consists of \(m\) clauses \(C_1, \ldots, C_m\) containing variables \(v_1, \ldots, v_n\). We may assume it satisfies the properties listed in Lemma 16. Let \(I_j\) denote the set of indices \(j\) such that \(v_j\) appears in \(C_j\). Let \(I_i\) denote the indices \(j\) such that \(\overline{v_j}\) appears in \(C_j\).

For the instance \(I\), we construct the graph \(G_I\) as follows. First, we create a complete graph on vertices \(y_1, \ldots, y_m\). Then for every occurrence of a variable \(v_i\) (resp. \(\overline{v_i}\)) in a clause \(C_j\), we add a new vertex \(x_{i,j}\) (resp. \(\overline{x}_{i,j}\)) and we add an edge between \(y_j\) and \(x_{i,j}\) (resp. \(\overline{x}_{i,j}\)). Finally, we add an edge between \(x_{i,j}\) and \(\overline{x}_{i,j}\) for all \(i \in \{1, \ldots, n\}\), all \(j \in I_i\), and all \(\ell \in I_i\).

We prove that \(G_I\) admits a \(\text{STABLE-M}_4^3\) partition if and only if \(I\) has a satisfying truth assignment (as an instance of \(\text{ONE-IN-THREE-3SAT}\)).

Suppose that the instance \(I\) has a satisfying truth assignment \(\varphi\). In other words, \(\varphi\) is a mapping from \(\{v_1, \ldots, v_n\}\) to \{true, false\} such that for every clause \(C_j\), \(\varphi\) evaluates exactly one of the literals in \(C_j\) to true, where \(\varphi(\overline{v_i})\) is defined as the negation of \(\varphi(v_i)\). Let us define a partition of \(V(G_I)\) as follows:
V_1 = \{ x_{i,j} \mid j \in I_i \land \varphi(v_i) = \text{false} \} \cup \{ \overline{x_{i,j}} \mid j \in \overline{I_i} \land \varphi(v_i) = \text{true} \},
V_2 = \{ x_{i,j} \mid j \in I_i \land \varphi(v_i) = \text{true} \} \cup \{ \overline{x_{i,j}} \mid j \in \overline{I_i} \land \varphi(v_i) = \text{false} \},
V_3 = \{ y_j \mid j \in \{ 1, \ldots, m \} \}.

It is not difficult to verify that V_1 and V_2 are independent sets of G_T, that V_3 is a clique, and that the edges between V_2 and V_3 form a perfect matching. Indeed, each vertex x_{i,j} or \overline{x_{i,j}} in V_2, namely the one for which v_i, resp. \overline{v}_i is the literal of C_j that \varphi evaluates to true. Thus G_T admits a \text{STABLE-M}_3\text{\#} partition as required.

Conversely, suppose that G_T admits a \text{STABLE-M}_3\text{\#} partition. In other words, there exists a partition of V(G_T) into three sets V_1, V_2, V_3 such that V_1 and V_2 are independent sets, V_3 is a clique, and the edges between V_2 and V_3 form a perfect matching.

First, we show that we must have V_3 = \{ y_j \mid j \in \{ 1, \ldots, m \} \}. By Lemma 16, there are at least four y_j's. Thus, since V_1 and V_2 are independent sets, V_3 must contain at least two y_j's. This implies that V_3 contains no x_{i,j} or \overline{x_{i,j}}, since each has at most one neighbour in \{ y_1, \ldots, y_m \} and V_3 is a clique. It also implies that if y_j \in V_2 for some j, then y_j has at least 2 neighbours in V_3, which is a contradiction. Finally, suppose that y_j \in V_1 for some j. Consider a neighbour z \in \{ y_1, \ldots, y_m \} of y_j. (Note that z is x_{i,j} or \overline{x_{i,j}} for some i and there are exactly three such vertices.) Then z is not in V_3, since V_3 contains no x_{i,j} or \overline{x_{i,j}}. Also, z cannot be in V_1, since V_3 is independent. Thus z must be in V_2. But z has a unique neighbour in \{ y_1, \ldots, y_m \}, namely y_j, and hence, z does not have a neighbour in V_3, a contradiction. This proves that V_3 = \{ y_1, \ldots, y_m \}.

Now, we define the following truth assignment \varphi : \{ v_1, \ldots, v_n \} \rightarrow \{ \text{true, false} \}. For each i \in \{ 1, \ldots, n \}, we set \varphi(v_i) = \text{true} if x_{i,j} \in V_2 for some j, and set \varphi(v_i) = \text{false} otherwise. We prove that \varphi is a satisfying truth assignment for the instance I, which will conclude the proof.

Using the assignment \varphi, we prove that

V_1 = \{ x_{i,j} \mid j \in I_i \land \varphi(v_i) = \text{false} \} \cup \{ \overline{x_{i,j}} \mid j \in \overline{I_i} \land \varphi(v_i) = \text{true} \},
V_2 = \{ x_{i,j} \mid j \in I_i \land \varphi(v_i) = \text{true} \} \cup \{ \overline{x_{i,j}} \mid j \in \overline{I_i} \land \varphi(v_i) = \text{false} \}.

To show this, recall that for each i \in \{ 1, \ldots, n \}, every x_{i,j} is adjacent to every \overline{x_{i,j}} where j \in I_i and \ell \in \overline{I_i}. Thus if \varphi(v_i) = \text{true}, then x_{i,j} \in V_2 for some j which implies \overline{x_{i,j}} \in V_1 for all \ell \in \overline{I_i}, since V_2 is an independent set. Therefore, x_{i,j} \in V_2 for all j \in I_i, since V_1 is an independent set. Similarly, if \varphi(v_i) = \text{false}, then x_{i,j} \in V_2 for all j \in I_i, and hence, \overline{x_{i,j}} \in V_2 for all \ell \in \overline{I_i}.

Now, consider a clause C_j. Recall that y_j \in V_3, and hence, it has exactly one neighbour x_{i,j} or \overline{x_{i,j}} in V_2 corresponding to the literal v_i, resp. \overline{v}_i in C_j, which \varphi evaluates to \text{true} by the above. So, all other neighbours x_{i,j} or \overline{x_{i,j}} of y_j belong to V_1 and thus correspond to literals v_y, resp. \overline{v}_y which \varphi evaluates to false. This proves that C_j is satisfied by \varphi, and thus, proves that \varphi is a satisfying truth assignment.

This concludes the proof. \square

Similar constructions also work for the following two cases.

**Theorem 18.** The \text{STABLE-M}_3\text{\#} problem is NP-complete.

**Proof.** Again, we rephrase the problem as: given a graph G, decide whether the vertices of G can be partitioned into 3 sets V_1, V_2, V_3 such that V_3 is a clique, V_1 and V_2 are independent sets and the edges between V_2 and V_3 form the complement of a perfect matching.

The proof will now follow essentially the same steps as the proof of Theorem 17. We proceed by reduction from \text{ONE-IN-THREE-3SAT}.

Consider an instance I of the problem, namely the instance consists of m clauses C_1, \ldots, C_m containing variables v_1, \ldots, v_n. Again, we may assume it satisfies the properties listed in Lemma 16. We define J_i to be the set of indices j such that v_i appears in C_j, and define \overline{J_i} to be the set of indices j such that \overline{v}_i appears in C_j.
For the instance $\mathcal{I}$, consider the graph $G^+_I$ constructed in the proof of Theorem 17. Let $G^+_I$ be the graph constructed from $G_I$ by complementing the edges between $\{y_1, \ldots, y_m\}$ and the rest of the graph. Namely, for each $i \in \{1, \ldots, m\}$, the vertex $y_i$ is adjacent to $z \notin \{y_1, \ldots, y_m\}$ in $G^+_I$ if and only if $y_i$ is not adjacent to $z$ in $G_I$. All other edges remain the same.

We prove that $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition if and only if $\mathcal{I}$ has a satisfying truth assignment (as an instance of $\text{ONE-IN-THREE-3SAT}$).

For the forward direction, we note that, by the proof of Theorem 17, if $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition $V_1, V_2, V_3$, then $V_3 = \{y_1, \ldots, y_m\}$. Thus, this is also a $\text{STABLE-}M^2_4$ partition of $G^+_I$. This proves that if $\mathcal{I}$ has a satisfying truth assignment, then $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition.

Conversely, suppose that $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition. Namely, let $V_1, V_2, V_3$ be a partition of $V(G_I)$ such that $V_1, V_2$ are independent sets, $V_3$ is a clique, and the edges between $V_2$ and $V_3$ form the complement of a perfect matching.

We shall prove that $V_3 = \{y_1, \ldots, y_m\}$. By the construction of $G^+_I$, this will imply that $V_1, V_2, V_3$ is also a $\text{STABLE-}M^2_4$ partition of $G_I$. Thus, by the proof of Theorem 17, this will allow us to conclude that $\mathcal{I}$ has a satisfying truth assignment.

Consider a vertex $y_i$. By Lemma 16, there is a variable $v_i$ such that neither $v_i$ nor $\overline{v_i}$ appears in the clause $C_j$. Moreover, $v_i$ appears as a literal at least twice, say $C_{j_1}$ and $C_{j_2}$, and $\overline{v_i}$ appears in two other clauses, say $C_{j_3}$ and $C_{j_4}$. This implies that $G^+_I$ contains vertices $x_{i,j_1}, x_{i,j_2}, \overline{x_{i,j_1}}, \overline{x_{i,j_2}}$ which induce a 4-cycle and are all adjacent to $y_i$. Suppose that $y_i \in V_1$. Since $V_1$ is an independent set, we conclude that $x_{i,j_1}, x_{i,j_2}, \overline{x_{i,j_1}}, \overline{x_{i,j_2}} \in V_2 \cup V_3$. However, this contradicts the fact that $G^+_I[V_2 \cup V_3]$ is a split graph. Thus $y_i \notin V_1$. By the same argument, $y_i \notin V_2$. This proves that $V_3 \supseteq \{y_1, \ldots, y_m\}$. Furthermore, note that $V_3$ contains no $x_{i,j}$ or $\overline{x_{i,j}}$ since each has a non-neighbour in $\{y_1, \ldots, y_m\}$ and $V_3$ is a clique. So $V_3 = \{y_1, \ldots, y_m\}$ as promised.

This concludes the proof.

\begin{theorem}
The $\text{STABLE-}M^2_2$ problem is NP-complete.
\end{theorem}

\begin{proof}
Once again we rephrase the problem as: given a graph $G$, decide if we can partition its vertex set into 3 independent sets $V_1, V_2, V_3$, such that the edges between $V_2$ and $V_3$ form the complement of a perfect matching. As before, we reduce from $\text{ONE-IN-THREE-3SAT}$.

Consider an instance $\mathcal{I}$ of the problem, namely the instance consists of $m$ clauses $C_1, \ldots, C_m$ containing variables $v_1, \ldots, v_n$. Again, we may assume it satisfies the properties listed in Lemma 16. We define $j_i$ to be the set of indices $j$ such that $v_i$ appears in $C_j$, and define $\overline{j_i}$ to be the set of indices $j$ such that $\overline{v_i}$ appears in $C_j$.

For the instance $\mathcal{I}$, consider the graph $G^+_I$ constructed in the proof of Theorem 18. Construct the graph $G^+_I$ from $G^+_I$ by removing all edges of the form $y_j, y_j$ where $i, j \in \{1, \ldots, m\}$ (effectively replacing the clique on $\{y_1, \ldots, y_m\}$ by an independent set). All other edges remain the same.

We claim that $G^+_I$ has a $\text{STABLE-}M^2_2$ partition if and only if $\mathcal{I}$ has a satisfying truth assignment (as an instance of $\text{ONE-IN-THREE-3SAT}$).

For the forward direction, we note that, by the proof of Theorem 18, if $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition $V_1, V_2, V_3$, then $V_3 = \{y_1, \ldots, y_m\}$. Thus, this is also a $\text{STABLE-}M^2_4$ partition of $G^+_I$. This proves that if $\mathcal{I}$ has a satisfying truth assignment, then $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition.

Now suppose, conversely, that $G^+_I$ admits a $\text{STABLE-}M^2_4$ partition. In other words, $V(G^+_I)$ can be partitioned into three independent sets $V_1, V_2, V_3$, such that the edges between $V_2$ and $V_3$ form the complement of a perfect matching.

First, observe that if three vertices $a, b, c \in V_2 \cup V_3$ form an independent set then either all of them must be contained in $V_2$ or all of them must be contained in $V_3$. Indeed, suppose, without loss of generality that, $a, b \in V_2$ and $c \in V_3$, then $c$ would have two non-neighbours in $V_2$, contradicting the fact that the edges between $V_2$ and $V_3$ form the complement of a perfect matching.

Next, we show that $y_j \in V_2$ for all $j \in \{1, \ldots, m\}$ or $y_j \in V_3$ for all $j \in \{1, \ldots, m\}$. By the above observation and Lemma 16, we need only show that $y_j \notin V_1$. Suppose, for contradiction, that $y_j \notin V_1$. By
Lemma 16, there must be vertices  \( x_{1,j_1}, x_{2,j_2}, x_{3,j_3} \) and  \( \overline{x_{1,i_4}} \) (with  \( i_1, i_2, i_3 \) pairwise distinct), none of which correspond to literals in the clause  \( C_j \) (i.e.  \( j \notin \{i_1, i_2, i_3, i_4\} \)). Since they do not correspond to these literals,  \( y_j \) must be adjacent to all of these vertices, so  \( x_{1,j_1}, x_{2,j_2}, x_{3,j_3}, \overline{x_{1,i_4}} \in V_2 \cup V_3 \). But  \( x_{1,j_1}, x_{2,j_2}, x_{3,j_3} \) and  \( \overline{x_{1,i_4}} \) are both independent sets of size 3. Thus all four of these vertices must be members of the same set  \( V_i \) where  \( i \in \{2, 3\} \). But  \( x_{1,j_1} \) and  \( \overline{x_{1,i_4}} \) are adjacent, contradicting the fact that  \( V_2 \) and  \( V_3 \) are independent sets.

Hence, we may conclude, without loss of generality, that  \( \{y_1, \ldots, y_m\} \subseteq V_3 \). Notice that, since each vertex  \( x_{1,j} \) or  \( \overline{x_{1,i}} \) corresponds to a unique occurrence of a literal in a unique clause in  \( \mathcal{I} \), every vertex not of the form  \( y_j \) has a neighbour in  \( V_3 \). Thus, since  \( V_3 \) is an independent set,  \( V_3 = \{y_1, \ldots, y_m\} \). Finally, note that since  \( V_1, V_2, V_3 \) is a  \( \text{STABLE-M}_2^3 \) partition for  \( G^+_z \) and  \( V_3 = \{y_1, \ldots, y_m\} \), then by the construction of  \( G^+_z \), it follows that  \( V_1, V_2, V_3 \) must also be a  \( \text{STABLE-M}_4^3 \)-partition of  \( G^+_z \). Thus, by the proof of Theorem 18, the instance  \( \mathcal{I} \) has a satisfying assignment.

This concludes the proof. \( \square \)

6. Conclusion

We proved that the  \( \text{STABLE-II} \) problem is polynomial-time solvable for all subfactorial hereditary properties  \( \Pi \) and for seven of the nine minimal factorial hereditary properties. For  \( \Pi = \mathcal{M}_1 \), the problem is known to be NP-complete. This leaves one final open case, namely where  \( \Pi \) is the class of chain graphs  \( \mathcal{M}_3 \). Clarifying the complexity status of this exception is a challenging research problem.

Acknowledgements

Research supported by the Centre for Discrete Mathematics and its Applications (DIMAP), which is partially funded by EPSRC award EP/D063191/1. The second and third author also acknowledge support from EPSRC award EP/I01795X/1.

References