Stable Matching Games: Manipulation via Subgraph Isomorphism

Sushmita Gupta[1] and Sanjukta Roy[2]

1 **Institute for Informatics, University of Bergen, Norway**
Sushmita.Gupta@uib.no, sushmita.gupta@gmail.com

2 **The Institute of Mathematical Sciences, HBNI, Chennai, India**
sanjukta@imsc.res.in

Abstract

In this paper we consider a problem that arises from a strategic issue in the stable matching model (with complete preference lists) from the viewpoint of exact-exponential time algorithms. Specifically, we study the Stable Extension of Partial Matching (SEOPM) problem, where the input consists of the complete preference lists of men, and a partial matching. The objective is to find (if one exists) a set of preference lists of women, such that the men-optimal Gale Shapley algorithm outputs a perfect matching that *contains* the given partial matching. Kobayashi and Matsui [*Algorithmica, 2010*] proved this problem is NP-complete. In this article, we give an exact-exponential algorithm for SEOPM running in time $2^{\mathcal{O}(n)}$, where n denotes the number of men/women. We complement our algorithmic finding by showing that unless Exponential Time Hypothesis (ETH) fails, our algorithm is asymptotically optimal. That is, unless ETH fails, there is no algorithm for SEOPM running in time $2^{o(n)}$. Our algorithm is a non-trivial combination of a parameterized algorithm for Subgraph Isomorphism, a relationship between stable matching and finding an out-branching in an appropriate graph and enumerating non-isomorphic out-branchings.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases stable matching, Gale-Shapley algorithm, suitor graph, subgraph isomorphism, exact-exponential time algorithms

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.29

1 Introduction

Stable Matching together with its in numerous variants are among the most well-studied problems in matching theory, driven by applications to economics, business, engineering, and more recently medical sciences. In the two-sided Stable Matching problem (also called the Stable Marriage problem), we are given two sets of agents of equal size, known as *men* and *women*, where each person submits a ranked list of all the members of the opposite sex. In this setting, a *matching* is a set of man-woman pairs (called matching partners), no two of which share a common member. A **stable matching** is a matching for which there does not exist a *blocking pair*: a man and a woman, who are not part of a matching pair, but prefer each other to their respective matching partners.

Ever since the theoretical framework for Stable Matching was laid down by Gale and Shapley [9] to study the then current heuristic used to assign medical residents to hospitals in New England, the topic has received considerable attention from theoreticians and practitioners alike. In particular, it is one of the foundational problems in social choice theory, where a matching is viewed as an *allocation* or *assignment* of resources to

36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016).
Editors: Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen; Article No. 29; pp. 29:1–29:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

relevant agents, whereby the nature of the assignment can vary greatly depending on the scenario/marketplace they are modelling. We refer the reader to books [10, 18, 14] for an in-depth introduction to stable matching and its variants.

Gale and Shapley [9] showed that every instance of the Stable Matching problem admits a stable matching. In other words, given any set of preference lists of men and women there exists at least one stable matching. In fact, they gave a polynomial time algorithm to find a stable matching. This algorithm is widely used in both practice and theory, and it exists in two versions: the *men-optimal* and the *women-optimal*, so named to emphasise the fact that one side prefers one over the other. Both variants are defined analogously. As the name suggests, the men-optimal stable matching is a stable matching that is no worse than any other stable matching, in terms of the preferences of the men. In other words, there does not exist a stable matching such that each man prefers his partner in that matching to his partner in the men-optimal stable matching. The algorithm that yields the men-optimal stable matching is called the *men-proposing* (resp. *women-proposing*) Gale-Shapley algorithm. The men-proposing version of the algorithm works as follows. A man who is not yet matched to a woman, *proposes* to the woman who is at the top of his current list, which is obtained by removing from his original preference list, all the women who have rejected him at an earlier step. On the woman's side, when a woman w receives a proposal from a man m, she accepts the proposal if it is her first proposal, or if she prefers m to her current partner. If w prefers her current partner to m, then w *rejects* m. If m is rejected by w, then m removes w from his list. This process continues until there is no unmatched man. The output of this algorithm is the men-optimal stable matching. For more details, see [10]. It has been customary to use the men-proposing version of the algorithm, and our analysis here will stick to that convention. Henceforth, unless explicitly stated otherwise, any mention of a stable matching should be interpreted by the reader as such. We will use $(\mathcal{L}^M, \mathcal{L}^W)$ to denote the set of preference lists of men and women, and the men-optimal matching with respect to these lists is denoted by $\mathrm{GS}(\mathcal{L}^M, \mathcal{L}^W)$.

1.1 Our problem and motivation

Kobayashi and Matsui [12, 13] studied manipulation in the stable matching model, where agents are manipulating with the goal of attaining a specific matching target. Formally speaking, they considered the following class of problems. An input consists of two sets M and W, (each of size n) of *men* and *women*, respectively; along with the preference list of every man (expressed as a strict ordering on the set of women) (denoted by $\mathcal{L}^M$) and a matching on (M, W). The said matching can either be *perfect* (if it contains n pairs), or *partial* (possibly, fewer than n pairs). Furthermore, for a couple of problems, we are given a set of preference lists of women, $\mathcal{L}^{W'}$, where $W' \subseteq W$. The goal is to decide if there exists a set of preference lists of women, $\mathcal{L}^W$, containing $\mathcal{L}^{W'}$, such that when used in conjunction with $\mathcal{L}^M$ with the men-optimal stable matching algorithm, yields a matching that contains all the pairs in the stated matching. Of these problems, two are directly related to our work in this paper. Let us consider the following two problems, and compare and contrast their computational complexity.

Attainable Stable Matching (ASM)
Input: A set of preference lists $\mathcal{L}^M$ of men over women W, and a perfect matching μ on (M, W).
Question: Does there exist a set of preference lists of women $\mathcal{L}^W$, such that $\mathrm{GS}(\mathcal{L}^M, \mathcal{L}^W) = \mu$?

Kobayashi and Matsui in [12, 13] showed that ASM is polynomial time solvable, and exhibited an $\mathcal{O}(n^2)$ algorithm that computes the set $\mathcal{L}^W$, if it exists. Or else, reports "none exists". The following problem is identical to the above, except in one key aspect: *the target matching need not be perfect.* The authors show that this problem is NP-complete.

> Stable Extension of Partial Matching (SEOPM)
> **Input:** A set of preference lists $\mathcal{L}^M$ of men M over women W, and a partial matching μ' on (M, W).
> **Question:** Does there exist preferences of women $\mathcal{L}^W$, such that $\mu' \subseteq \text{GS}(\mathcal{L}^M, \mathcal{L}^W)$?

These two problems and their differing computational complexities represent a dichotomy with respect to the size of target matching. Kobayashi and Matsui solve ASM by designing a novel combinatorial structure called the *suitor graph*, which encodes enough information about the men's preferences and the matching pairs, that it allows an efficient search of the possible preference lists of women, which are $n \cdot n!$ in number. The same approach falls short when the stated matching is partial.

Our work in this paper falls thematically within the area of strategic results relating to the stable matching problem. There is a long history of results centred around the question as to whether an individual agent, or a coalition of agents can misstate their true preference lists (either by truncating, or by permuting the list), with the objective of obtaining a better partner (assessed in terms of the true preferences of the manipulating agents) than would otherwise be possible under the men-optimal stable matching algorithm. SEOPM is to be viewed as a manipulation game in which a coalition of agents (in this case the subset of women who are matched under the partial matching) have decided upon a specific partner. These agents are colluding, with co-operation from the other women who are not matched, to produce a perfect matching, which gives each of the manipulating agents their target partners. There exists a strategy to attain this objective if and only if there exists a set of preference list of women that yields a perfect matching that contains the partial matching.

Since SEOPM has been shown to be NP-complete, it is natural to study this problem in computational paradigms that are meant to cope with NP-hardness. We attempt such a study in the area of exact exponential time algorithms. Manipulation and strategic issues in voting have been well-studied in the field of exact algorithms and parameterized complexity; see the survey [3] for an overview. But one can not say the same regarding the strategic issues in the stable matching model. These problems hold a lot of promise and remain hitherto unexplored in the light of exact algorithms and parameterized complexity, with exceptions that are few and far between [15, 16].

To the best of our knowledge, Cseh and Manlove [4] initiated this type of analysis by studying an NP-hard variant of the stable marriage and *stable roommate* problems[1], where the input consists of each of the preference lists, as well two subsets of (not necessarily pairwise disjoint) pairs of agents, representing the *forbidden pairs* and the *forced pairs*. The goal is to find a matching that does not contain any of the forbidden pairs, and contains each of the forced pairs, while simultaneously minimizing the number of blocking pairs.

[1] In the stable roommate problem, the matching market consists of agents of the same type, as opposed to the market modelled the stable marriage problem that consists of agents of two types, men and women. Roommate assignments in college housing facilities is a real world application of the stable roommate problem.

1.2 Our Contributions

Throughout the article, n is used to denote $n = |M| = |W|$. The most basic algorithm for SEOPM would be to guess the permutation of all women (that is, the set of preferences of women, $\mathcal{L}^W$) and check whether $\mu' \subseteq \text{GS}(\mathcal{L}^M, \mathcal{L}^W)$. However, this algorithm will take $(n!)^n n^2 = 2^{\mathcal{O}(n^2 \log n)}$. One can obtain an improvement over this naïve algorithm by using the polynomial time algorithm for ASM [13]. That is, using the algorithm for ASM, which given a matching μ can check in polynomial time whether there exists $\mathcal{L}^W$ such that $\mu = \text{GS}(\mathcal{L}^M, \mathcal{L}^W)$. The faster algorithm for SEOPM, using the algorithm for ASM, tries all possible extensions of the partial matching $\mu \supseteq \mu'$ and checks in polynomial time whether there exists $\mathcal{L}^W$ such that $\mu = \text{GS}(\mathcal{L}^M, \mathcal{L}^W)$. Thus, if the size of the partial matching is k, this algorithm would have to try $(n-k)!$ possibilities. In the worst case this can take $(n!)n^{\mathcal{O}(1)} = 2^{\mathcal{O}(n \log n)}$.

In this article we give a $2^{\mathcal{O}(n)}$ algorithm, which not only breaks the naïve bound, but also uses an idea which connects SEOPM to the problem of Colored Subgraph Isomorphism (given two graphs G and H, the objective is to test whether H is isomorphic to some subgraph of G). We establish this connection by introducing a combinatorial tool, the *universal suitor graph* that extends the notion of the rooted *suitor graph* devised by Kobayashi and Matsui in [12, 13], to solve ASM. It is shown in [13] that an input instance $(\mathcal{L}^M, \mu)$ of ASM is a Yes-instance if and only if the corresponding rooted suitor graph has an *out-branching* : a spanning subgraph in which every vertex has at most one in-coming arc, and is reachable from the root. The universal suitor graph satisfies the property that $(\mathcal{L}^M, \mu')$, an instance of SEOPM is a Yes-instance if and only if the corresponding universal suitor graph contains a subgraph that is isomorphic to the out-branching corresponding to $(\mathcal{L}^M, \mu)$ where μ is the perfect matching that "extends" μ'. Thus, the universal suitor graph succinctly encodes all "possible suitor graphs" and is only polynomially larger than the size of a suitor graph. That is, the size of universal suitor graph is $\mathcal{O}(n^2)$. This is our main conceptual contribution and we believe that the concept of the universal suitor graph is likely to be of independent interests, useful in characterizing existence of strategies in other manipulation games.

Using ideas from exact exponential algorithms and parameterized complexity; in particular by using as a subroutine the algorithm that enumerates all non-isomorphic out-branchings in a (given) rooted directed graph [2, 17], and a parameterized algorithm for Colored Subgraph Isomorphism [1, 7, 8], we can search for a subgraph in the universal suitor graph that is isomorphic to an out-branching corresponding to an extension of μ'. We complement our algorithmic finding by showing that unless Exponential Time Hypothesis (ETH) fails, our algorithm is asymptotically optimal. That is, unless ETH fails, there is no algorithm for SEOPM running in time $2^{o(n)}$. We refer to the following books for further reading regarding exact algorithms [6] and parameterized complexity [5].

2 Preliminaries

For a positive integer n, we will use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. As introduced earlier, M and W denote the set of men and women, respectively, and we assume that $|M| = |W| = n$. Each $m \in M$ has a preference list, denoted by $P(m)$, which is a total ordering of W. The set of preference lists of all men is denoted by $\mathcal{L}^M$. Similarly, each $w \in W$ has a preference list, denoted by $P(w)$ which is a total ordering of M. The set of preference lists of all women is denoted by $\mathcal{L}^W$. It is helpful to view (M, W) as the bipartitions of a complete bipartite graph, and a perfect matching in (M, W) as a set of vertex disjoint edges that matches every vertex in $M \cup W$. Similarly, a partial matching in (M, W) can be viewed as a set of vertex disjoint edges that does not necessarily match every vertex in $M \cup W$.

Given a matching μ (perfect or partial), and a vertex $v \in M \cup W$, $\mu(v)$ denotes the matched partner of the man/woman v. We note that for a perfect matching μ: $m \in M$ if and only if $\mu(m) \in W$, and similarly $w \in W$ if and only if $\mu(w) \in M$. But, when we have a partial matching, μ, it may be that some vertices (male or female) are not matched under it, we denote that symbolically as $\mu(v) = v$ for any man/woman $v \in M \cup W$ who is not matched in μ. A matching μ is said to be **an extension** of a matching μ' if $\mu' \subseteq \mu$, that is μ contains the set of edges in μ'. For any matching μ, and a man m matched in μ, we define $\delta^+(m) = \{w \in W | \, m \text{ strictly prefers } w \text{ to } \mu(m)\}$, and conversely for any woman $w \in W$ (not necessarily matched in μ) we define $\delta^-(w) = \{m \in M | \, m \text{ strictly prefers } w \text{ to } \mu(m)\}$; all preferences are in terms of lists in $\mathcal{L}^M$.

Throughout the paper, we use the standard notations about directed graphs. Given a directed graph D, and a vertex $v \in V(D)$, we use $N^-(v)$ to denote the set of vertices that are in-neighbors of v: $N^-(v) = \{u \mid (u,v) \in E(D)\}$. Similarly, we use $N^+(v)$ to denote the set of vertices that are out-neighbors of v: $N^+(v) = \{u \mid (v,u) \in E(D)\}$. Following the usual notations, a source is a vertex v such that $N^-(v) = \emptyset$ and a sink is a vertex v such that $N^+(v) = \emptyset$. An **out-branching** is a directed graph with a special vertex, called the *root*, where each vertex is reachable from the root by exactly one directed path. Essentially, this is a rooted tree with all arcs oriented away from the root. For any directed edge or an arc, *tail* is the vertex from where the arc originates and the *head* is the vertex at which it ends.

3 Generalization of Suitor Graph

The main tool we use to obtain our exact exponential time algorithm is the notion of a *universal suitor graph* – a generalization of the *suitor graph* introduced by Kobayashi and Matsui [13]. We start the section by introducing the definition of a suitor graph, followed by the definition of a universal suitor graph.

Suitor Graph and Rooted Suitor Graph. Given a set of preference lists $\mathcal{L}^M$ of men over set of women W and a partial matching μ', $G(\mathcal{L}^M, \mu')$ denotes a directed bipartite graph, called a *suitor graph*, where $V(G) = M \cup W$ and a set of directed arcs $E(G)$ defined as follows,

$$E(G) = \Big\{(w, \mu'(w)) \in W \times M \ \mid \ w \text{ is matched in } \mu'\Big\} \cup \Big\{(m, w) \in M \times W \mid m \text{ is matched in } \mu', w \in \delta^+(m)\Big\}.$$

Observe that the arcs for which a woman is the tail are the (only) arcs that correspond to the matched pairs in μ'.

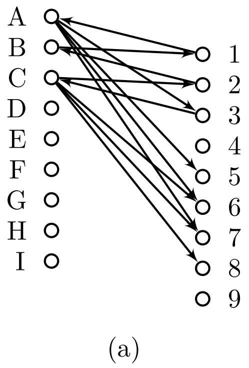
For a given suitor graph $G(\mathcal{L}^M, \mu')$, the associated *rooted suitor graph* is a directed graph $\overline{G}(\mathcal{L}^M, \mu')$ defined as follows. We introduce an artificial vertex r, called the *root*, to $G(\mathcal{L}^M, \mu')$ and add arcs (r, w) for every vertex $w \in W$ that has no incoming arcs in $G(\mathcal{L}^M, \mu')$. That is, we add arcs from r to all the vertices that are *sources* in $G(\mathcal{L}^M, \mu')$. We give an example of a suitor graph and a rooted suitor graph. Figure 1 shows the suitor graph and the rooted suitor graph for the preference lists given in Table 1 and the partial matching $\{(A, 1), (B, 2), (C, 3)\}$. The vertex marked as r is the root vertex.

Our main motivation for suitor graph and its generalization is the following result proved in [13, Theorem 2].

▶ **Proposition 1** ([13]). *Let $\mathcal{L}^M$ be a set of preference lists for M, and μ be a perfect matching between (M, W), then the following holds. There exists $\mathcal{L}^W$, a set of preference lists for*

Table 1 Example: Preference List of Men over Women.

Man	Preference over women								
A	3	7	6	5	1	9	8	4	2
B	1	2	4	3	9	5	8	7	6
C	2	7	6	8	3	4	9	1	5
D	2	7	6	8	3	4	9	1	5
E	3	7	6	5	1	9	8	4	2
F	1	2	4	3	9	5	8	7	6
G	3	7	6	5	1	9	8	4	2
H	1	2	4	3	9	5	8	7	6
I	2	7	6	8	3	4	9	1	5



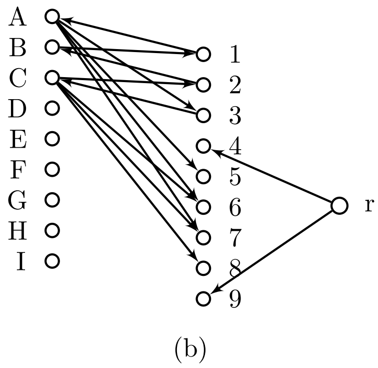


Figure 1 (a) Suitor Graph, (b) Rooted Suitor Graph.

W such that $\text{GS}(\mathcal{L}^M, \mathcal{L}^W) = \mu$ *if and only if the rooted suitor graph* $\overline{G}\left(\mathcal{L}^M, \mu\right)$ *has an out-branching.*

There exists a polynomial time algorithm that takes as input $(\mathcal{L}^M, \mu)$ and outputs $\mathcal{L}^W$ (if one exists) such that $\text{GS}(\mathcal{L}^M, \mathcal{L}^W) = \mu$. Otherwise, it reports "none exists". We will be using this as a subroutine in our algorithm which will be presented in a later section.

Universal Suitor Graph. Next we define *universal suitor graph (USG)*. The idea is to construct a graph that given a set of preference lists $\mathcal{L}^M$ of men over women captures all possible suitor graphs succinctly. Then we make use of this to solve our problem. Formally, given a set of preference lists $\mathcal{L}^M$ of men over women, universal suitor graph, $U(\mathcal{L}^M)$, is defined as follows. We make n different copies of each man $m_i \in M$, denoted by $M_i = \{m_i^1, \ldots, m_i^n\}$. Recall that for every $m_i \in M$, the preference list $P(m_i) \in \mathcal{L}^M$ is given. We define $P(m_i^j) = P(m_i)$, for $1 \leq j \leq n$. Thus, the vertex set of the graph is $V(U(\mathcal{L}^M)) = \biguplus_{i=1}^{n} M_i \cup W$. The arc set, $E(U(\mathcal{L}^M))$, is defined as follows. For every $w_i \in W$, the graph contains arcs (w_i, m_j^i) for all $1 \leq j \leq n$. Additionally, the graph contains the arc (m_i^j, w_k) if m_i prefers w_k to w_j in $P(m_i)$, $w_k, w_j \in W$. This condition is depicted notationally as $w_k >_{m_i} w_j$. The intuition behind the construction is the following: given any matching μ, if a man m_j is matched with woman w_k then we imitate that by matching w_k to the k^{th} copy of m_j. Furthermore, using other copies of m_j we imitate connections with women whom he prefers to w_k. In particular, the i^{th} copy of every man is "paired" to w_i, i.e., $N^+(w_i) = \{m_1^i, m_2^i, \ldots, m_n^i\}$. This idea of pairing is captured by the fact that every male vertex in USG (consider m_k^i) has a unique in-neighbor (the female vertex w_i).

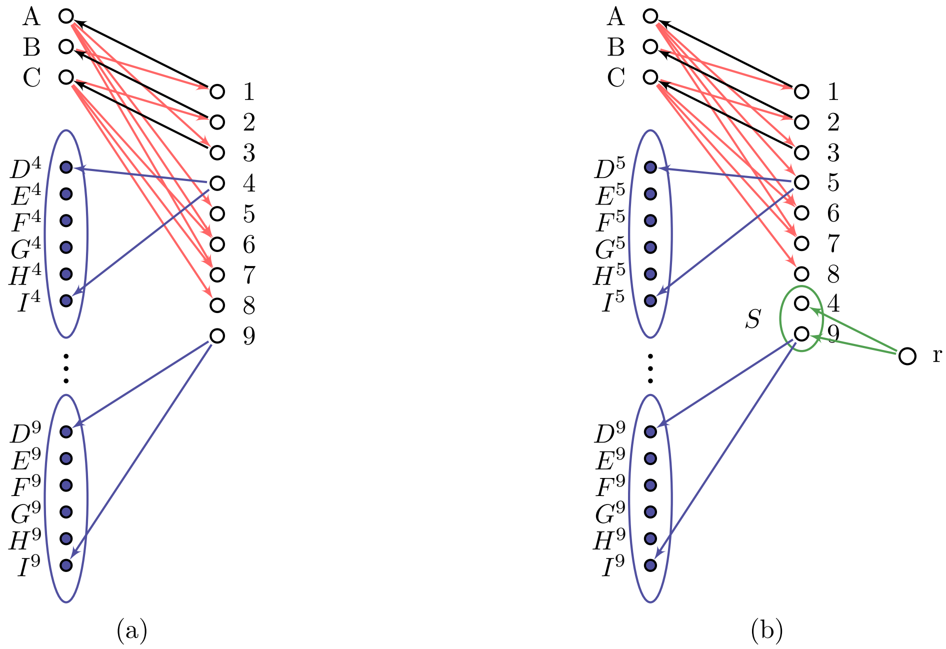


Figure 2 (a) Universal suitor graph, (b) Rooted universal suitor graph [described later in secion 4.2] for the partial matching $\mu = \{(A, 1), (B, 2), (C, 3)\}$ with sources in $\{4, 9\}$ (shows edges partially). Black edges represent the matching edges in μ, red edges represent the preferences of men matched in μ, while the blue edges represent edges from an unmatched woman to her own copies of the unmatched men. The green ellipse represents the set of source vertices, $\{4, 9\}$, that are connected from the root (not shown).

Universal Suitor Graph for a Partial Matching. For a given partial matching μ on the set (M, W), we define the graph, $U(\mathcal{L}^M, \mu)$, as follows. A man $m \in M$ is matched under μ if and only if $\mu(m) \in W$, and analogously for a woman $w \in W$, w is matched under μ if and only if $\mu(w) \in M$. We refer to the following set of operations collectively as the **pruning** of $U(\mathcal{L}^M)$ w.r.t. μ.

Matched Women: Let $\mu(w_i) = m_j$. Then delete vertices $\{m_k^i \mid 1 \le k \le n, k \ne j\}$, from the graph. This ensures that every matched female vertex w_i, has a unique out-going arc to the i^{th} copy of the man $\mu(w_i)$. In other words, only the arc (w_i, m_j^i), where $\mu(w_i) = m_j$, survives.

Unmatched Women: Let $\mu(w_i) \notin M$. Then delete vertices $\{m_k^i \mid 1 \le k \le n,\ \mu(m_k) \in W\}$. That is, delete the i^{th} copy of a man who is matched under μ. This ensures that in the subgraph, every unmatched female vertex w_i has out-going arcs to the vertices in the set $\{m_k^i \mid m_k \text{ is unmatched in } \mu\}$.

This completes the description of the pruning operations. Thus, to obtain the graph $U(\mathcal{L}^M, \mu)$ we start with $U(\mathcal{L}^M)$ and apply the pruning operations defined above with respect to the matching μ. Edges in $U(\mathcal{L}^M)$ that are not deleted during the above pruning operations, are said to have *survived pruning* w.r.t. μ. We give an example of a universal suitor graph for a partial matching. Figure 2 shows the universal suitor graph and the rooted universal suitor graph [described later in section 4.2] for the preference lists given in Table 1, and for the partial matching $\mu = \{(A, 1), (B, 2), (C, 3)\}$. To keep the figure clear, we only show the copies of male vertices for women 4 and 9. The edges going out of these copies of the male vertices are omitted.

We conclude this discussion with a useful lemma that will be invoked in several arguments.

▶ **Lemma 2.** *Let μ denote a partial matching. If a male vertex m_j^i survives pruning w.r.t. μ, then either $\mu(m_j) = w_i$, or else both m_j and w_i are unmatched in μ. Furthermore, the out-going arcs from m_j^i are also not deleted during pruning operations.*

Proof. We begin by noting that if w_i is matched to someone other that m_j, then the vertex m_j^i must be deleted during the pruning step; this is a contradiction. Suppose that w_i is unmatched, and $\mu(m_j) = w_\ell$. Then, the arc (w_ℓ, m_j^ℓ) survives, but the vertex m_j^i must be deleted, again a contradiction. Hence, the fact that m_j^i survives pruning w.r.t. μ, implies that either $\mu(m_j) = w_i$, or both m_j, w_i are unmatched.

Additionally, we note that if $\mu(m_j) = w_i$, then m_j^i is the sole member of M_j that survives the pruning steps. Also note that regardless of whether m_j is matched or unmatched, the out-going arcs from m_j^i survive the pruning process. ◀

4 Exact Algorithm for SEOPM

In this section we design a moderately exponential time algorithm for SEOPM. Towards this we will combine the following three ingredients:
- the notion of a universal suitor graph defined in the previous section;
- a parameterized algorithm for Subgraph Isomorphism when the pattern graph has bounded treewidth; and
- the fact that the number of non-isomorphic (i.e. unlabelled) trees on n vertices is at most $2.956^n n^{\mathcal{O}(1)}$.

We start this section by giving an overview of our algorithms. Towards this we first give the relevant notions and definitions.

▶ **Definition 3.** Two digraphs G_1 and G_2 are said to be isomorphic if there is a function $f : V(G_1) \to V(G_2)$ that satisfies the following properties:
1. f is a bijective function, i.e., f^{-1} is a function from $V(G_2)$ to $V(G_1)$;
2. for every edge $(u, v) \in E(G_1)$, we have $(f(u), f(v)) \in E(G_2)$.

A function such as f is called an *isomorphism function*. This function can be extended to sets of vertices analogously. That is, for all $V_1 \subseteq V(G_1), f(V_1) = \{f(v) \mid v \in V_1\} \subseteq V(G_2)$. We write $G_1 \simeq G_2$ to denote the two graphs are isomorphic.

Now we are ready to define the Colored Subgraph Isomorphism problem. The Colored Subgraph Isomorphism problem is formally defined as follows.

Colored Subgraph Isomorphism (Col-Sub-Iso) **Parameter:** $|V(H)|$
Input: A host graph G, a pattern graph H, and a coloring $\chi : V(G) \to \{1, 2, \ldots, |V(H)|\}$.
Question: Is there a subgraph G' in G such that $G' \simeq H$, and the vertices of G' have distinct colors?

We obtain the desired algorithm by making $2^{\mathcal{O}(n)}$ instances of the Col-Sub-Iso problem where the pattern graph has size $2n + 1$ and treewidth 3, and the given instance of SEOPM is a Yes instance if and only if one of the constructed instances is a Yes instance of the Col-Sub-Iso problem. Our host graph will be a universal suitor graph corresponding to an instance of SEOPM. We refer the reader to [5] for definitions of treewidth and tree decomposition. To solve Col-Sub-Iso we will use known algorithms, in particular, the algorithm alluded to in the following result.

▶ **Proposition 4** ([1]). *Let G and H denote two graphs on n and q vertices, respectively such that the treewidth of H is at most t. Furthermore, there is a coloring $\chi : V(G) \to [q]$ of G. Then there is a deterministic algorithm for* Col-Sub-Iso *that runs in time $2^q(nt)^{t+\mathcal{O}(1)}$, and outputs (if there exists one) a subgraph of G that has a distinct color on every vertex, and is isomorphic to H.*

To give the desired reduction to Col-Sub-Iso we essentially enumerate all non-isomorphic trees on $2n+1$ vertices. In the past, mainly rooted (undirected) trees have been studied, out-branchings not as much. However, every rooted tree can be made an out-branching by orienting every edge away from the root and every out-branching can be transformed into a rooted tree by disregarding all edge orientations. Thus, rooted trees and out-branchings are equivalent, and thus, the results obtained for the former are applicable to the latter. Otter [17] showed that the number of non-isomorphic out-branchings on n vertices is $t_n = 2.956^n n^{\mathcal{O}(1)}$. We can generate all non-isomorphic rooted trees on n vertices using the algorithm of Beyer and Hedetniemi [2] of runtime $\mathcal{O}(t_n)$. We summarize the above in the following result.

▶ **Proposition 5** ([2, 17]). *The number of non-isomorphic out-branchings on n vertices is $t_n = 2.956^n n^{\mathcal{O}(1)}$. Furthermore, we can enumerate all non-isomorphic rooted trees on n vertices in time $\mathcal{O}(t_n)$.*

4.1 Universality of Universal Suitor Graph

In this section we show the "universality" of the universal suitor graph. That is, how given a set of preference lists, $\mathcal{L}^M$, of men over women, universal suitor graph encodes all potential suitor graphs. Universal suitor graph for a partial matching encodes all suitor graphs of all potential extensions of the given partial matching. In particular, we show the following result.

▶ **Lemma 6.** *Let $\mathcal{L}^M$ denote a set of preference lists of men over women and let μ' denote a partial matching on the set (M, W). If there exists a perfect matching μ such that $\mu' \subseteq \mu$ (as a set of edges), then $U(\mathcal{L}^M, \mu)$ is a subgraph of $U(\mathcal{L}^M, \mu')$, and is isomorphic to the suitor graph $G\left(\mathcal{L}^M, \mu\right)$.*

Proof. Let M' and W' denote the subset of men and women who are matched under μ', respectively. Let $\mu' \subseteq \mu$, in terms of a subset of edges. We will refer to the suitor graphs $G\left(\mathcal{L}^M, \mu'\right)$ and $G\left(\mathcal{L}^M, \mu\right)$ as simply suitor graphs for μ' and μ, respectively.

Consider the universal suitor graph for μ, denoted by $U(\mathcal{L}^M, \mu)$, obtained from $U(\mathcal{L}^M)$ by pruning w.r.t. μ. Since $\mu' \subseteq \mu$ for every $w \in W'$ $(m \in M')$ we have $\mu(w) = \mu'(w)$ $(\mu(m) = \mu'(m))$. Thus, it is easy to see that $U(\mathcal{L}^M, \mu)$ is a subgraph of $U(\mathcal{L}^M, \mu')$, and can be obtained from the latter by applying the pruning operation to every female vertex $w_i \in W \setminus W'$. The next claim completes the proof, since it leads to the conclusion that the suitor graph $G\left(\mathcal{L}^M, \mu\right)$ is isomorphic to the universal suitor graph $U(\mathcal{L}^M, \mu)$.

▶ **Claim 7.** *Suitor graph $G\left(\mathcal{L}^M, \mu\right)$ is isomorphic to $U(\mathcal{L}^M, \mu)$.*

Proof. By the construction of $G\left(\mathcal{L}^M, \mu\right)$, we know the suitor graph of μ has arcs $(w, \mu(w))$ for every $w \in W$. Since $U(\mathcal{L}^M, \mu)$ is obtained from $U(\mathcal{L}^M)$ by pruning w.r.t. μ, hence we know that $U(\mathcal{L}^M, \mu)$ contains $2|\mu|$ vertices

$$\biguplus_{w_i \in W} \{w_i, m_j^i \mid \mu(w_i) = m_j\}.$$

We use M^μ to denote the male vertices in $U(\mathcal{L}^M, \mu)$.

Let $\Psi_\mu : M \cup W \to M^\mu \cup W$ denote a function between the vertex sets of $G\left(\mathcal{L}^M, \mu\right)$ and $U(\mathcal{L}^M, \mu)$. For every $w_i \in W$, we define $\Psi_\mu(w_i) = w_i$, and for every $m_i \in M$, we define $\Psi_\mu(m_i) = m_i^j$, where $\mu(m_i) = w_j$. We will prove that the map Ψ_μ is an isomorphism.

We begin with the observation that both graphs are bipartite, with vertex set (M, W) and (M^μ, W). Thus, to prove that Ψ_μ is an isomorphism, it is sufficient to prove that for every $w \in W, m \in M$, (w, m) is an arc in $G\left(\mathcal{L}^M, \mu\right)$ if and only if $(w, \Psi_\mu(m))$ is an arc in $U(\mathcal{L}^M, \mu)$, and similarly (m, w) is an arc in $G\left(\mathcal{L}^M, \mu\right)$ if and only if $(\Psi_\mu(m), w)$ is an arc in $U(\mathcal{L}^M, \mu)$.

Let (w_i, m_j) be an arc in $G\left(\mathcal{L}^M, \mu\right)$. Thus, we have $\mu(w_i) = m_j$, and so $\Psi_\mu(m_j) = m_j^i$. The construction of $U(\mathcal{L}^M, \mu)$ (that is pruning w.r.t. μ) ensures that (w_i, m_j^i) is an arc in $U(\mathcal{L}^M, \mu)$. Conversely, if (w_i, m_j^i) is an arc in $U(\mathcal{L}^M, \mu)$ then since μ is a perfect matching, by Lemma 2, we can conclude that $\mu(w_i) = m_j$, and so (w_i, m_j) is an arc in $G\left(\mathcal{L}^M, \mu\right)$. This completes the proof of the if and only if statement about female to male arcs.

Let (m_i, w_k) be an arc in $G\left(\mathcal{L}^M, \mu\right)$ i.e., $\mu(m_i) = w_j$. Thus, $w_k >_{m_i} w_j$ (m_i prefers w_k to w_j in $\mathcal{L}^M$). The vertex $m_i^j = \Psi_\mu(m_i)$ and the arc (m_i^j, w_k) exists in the universal suitor graph $U(\mathcal{L}^M)$. If we can show that m_i^j exists in $U(\mathcal{L}^M, \mu)$, then by the additional condition of Lemma 2, we know that the arc (m_i^j, w_k) exists in $U(\mathcal{L}^M, \mu)$. We note that m_i^j must survive the pruning of $U(\mathcal{L}^M)$ w.r.t. μ because (w_j, m_i) is an arc in $G\left(\mathcal{L}^M, \mu\right)$ and so from the earlier part we know that (w_j, m_i^j) is an arc in $U(\mathcal{L}^M, \mu)$. Hence, m_i^j must be a vertex in $U(\mathcal{L}^M, \mu)$, and so we conclude that $(\Psi_\mu(m_i), w_k)$ is an arc in $U(\mathcal{L}^M, \mu)$. Conversely, if (m_i^j, w_k) is an arc in $U(\mathcal{L}^M, \mu)$, then the presence of m_i^j in the graph allows us to invoke Lemma 2 to conclude that $\mu(m_i) = w_j$. This implies that $w_k >_{m_i} w_i$, hence (m_i, w_k) must also be an arc in $G\left(\mathcal{L}^M, \mu\right)$. This completes the proof of the if and only if statement about male to female arcs. Hence, our proof is complete. ◀

Since $U(\mathcal{L}^M, \mu)$ is a subgraph of $U(\mathcal{L}^M, \mu')$, hence by Claim 7 the latter contains a subgraph that is isomorphic to $G\left(\mathcal{L}^M, \mu\right)$. This completes the proof. ◀

4.2 Rooted Universal Suitor Graph and Valid Subgraphs

For a given universal suitor graph $U(\mathcal{L}^M, \mu')$ and a subset $S \subseteq W$, we define the corresponding **rooted universal suitor graph with sources in** S, as follows. For a vertex $w \in S$, if w is a source in $U(\mathcal{L}^M, \mu')$ (i.e. $N^-(w) = \emptyset$) then we add the arc (r, w). Otherwise, we delete all the male vertices in $N^-(w)$, and add the arc (r, w). The resulting graph is the rooted universal suitor graph with sources in S, and is denoted by $\overline{U}(\mathcal{L}^M, \mu', S)$. We refer the reader to Figure 2(b) for an example of a rooted universal suitor graph. The set of vertices marked as S is the set of source vertices that are connected to the root.

Recall that in a universal suitor graph for a partial matching there may be multiple copies of a male vertex, and that brings us to the notion of a valid subgraph. A subgraph of $U(\mathcal{L}^M, \mu')$ is said to be a **valid subgraph** if it contains every female vertex, and exactly one copy of every male vertex. The definition can be extended to the rooted subgraphs of $\overline{U}(\mathcal{L}^M, \mu', S)$, where $S \subseteq W$, and a valid rooted subgraph contains the root, every female vertex and exactly one copy of every male vertex.

Consider a rooted tree, such that the root is considered to be in layer 0. A vertex v is said to be in layer i in the tree, if the (unique) path from the root to v contains i arcs. A rooted tree is called a **matching tree** if every vertex in an odd layer has a unique child in the tree. If a matching tree is a valid subgraph of $U(\mathcal{L}^M, \mu')$ then it is called a **valid matching tree**. We note that a matching tree is also an out-branching.

Given a matching tree T, we construct the **triangular matching tree** $T^{\triangle}$, by adding two new vertices r^1 and r^2 to T and adding the arcs $(r, r^1), (r^1, r^2)$ and (r^2, r). Similarly, for any given rooted universal suitor graph $\overline{U}(\mathcal{L}^M, \mu', S)$, we construct the **triangular rooted universal suitor graph**, $\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S)$, by adding two new vertices r^1 and r^2 to T and adding the arcs $(r, r^1), (r^1, r^2)$ and (r^2, r).

Finally, we define the special coloring χ_{sp} used to color the vertices of a triangular rooted universal suitor graph: χ_{sp} uses $2n + 3$ colors, giving distinct colors to $r, r^1, r^2, w_1, \ldots, w_n$, and using the remaining n colors such that the subset of copies of the same male vertex gets a distinct color. That is, for each i $(1 \leq i \leq n)$ the subset of $\{m_i^1, \ldots, m_i^n\}$ that exists in the universal suitor graph gets the $n + 3 + i^{th}$ color.

4.3 $2^{\mathcal{O}(n)}$ Algorithm for SEOPM

In this section we combine all the results we have developed so far and design our algorithm.

Overview of Algorithm 4.1: Let $(\mathcal{L}^M, \mu')$ be an input instance of SEOPM. If μ' can be extended to μ, then (by Lemma 6), we know that $G(\mathcal{L}^M, \mu)$ is isomorphic to a subgraph in $U(\mathcal{L}^M, \mu')$. If μ' cannot be extended, then by Proposition 1 we know that for any perfect matching $\mu \supseteq \mu'$, the graph $\overline{G}(\mathcal{L}^M, \mu)$ does not contain an out-branching rooted at r. In other words, there exists a vertex v that is not reachable from r in the graph $\overline{G}(\mathcal{L}^M, \mu)$. Consequently, to "solve" SEOPM on $(\mathcal{L}^M, \mu')$, it is necessary and sufficient to look for a *valid out-branching or matching tree in the universal suitor graph* $U(\mathcal{L}^M, \mu')$. If the algorithm finds one, we can conclude that μ' can be extended, else it answers that μ' cannot be extended. We implement these ideas by constructing an appropriate instance of COL-SUB-ISO.

The algorithm works as follows. Assume that we have a stable matching μ that extends μ'. Then consider the graph $G(\mathcal{L}^M, \mu)$ and let S denote the subset of female vertices that are sources in the graph. Our algorithm implements this by enumerating all subsets S of W in the first loop. Furthermore, by Proposition 1 there is a matching tree, T, rooted at r in $\overline{G}(\mathcal{L}^M, \mu)$. To "guess" the tree T, we enumerate all non-isomorphic out-branchings on $2n+1$ vertices and first check whether it is a matching tree. If the enumerated tree is a matching tree then we create an instance of COL-SUB-ISO, where the host graph is $\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S)$, with its vertices colored by χ_{sp}, and the pattern graph is $T^{\triangle}$. Finally, using an algorithm for COL-SUB-ISO described in Proposition 4, we test whether, or not $(\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S), T^{\triangle}, \chi_{sp})$ is a YES-instance of COL-SUB-ISO. If the algorithm returns T^*, we can conclude that a stable matching μ extends μ'. If the outermost for-loop terminates without finding a YES-instance of COL-SUB-ISO, then we return that "no valid out-branching exists" (and hence no stable extension exists). This concludes the description of the algorithm. We refer the reader to Algorithm 4.1 for further details. The next lemma argues the correctness of Algorithm 4.1.

► **Lemma 8.** *Let $(\mathcal{L}^M, \mu')$ denote an input to* SEOPM. *Then $(\mathcal{L}^M, \mu')$ is a* YES*-instance of* SEOPM *if and only if Algorithm 4.1 returns a triangular matching tree T^*.*

Proof. Let $(\mathcal{L}^M, \mu')$ be a YES-instance, i.e., there exists a perfect matching μ, such that $\mu' \subseteq \mu$, and there exists $\mathcal{L}^W$ such that $\mu = \mathrm{GS}(\mathcal{L}^M, \mathcal{L}^W)$. By Lemma 6, $G(\mathcal{L}^M, \mu)$ is isomorphic to a subgraph in $U(\mathcal{L}^M, \mu')$.

By Proposition 1 $\overline{G}(\mathcal{L}^M, \mu)$ has an out-branching rooted at r, denoted by $\widetilde{T}$. Since by Lemma 6 $G(\mathcal{L}^M, \mu)$ is isomorphic to a subgraph in $U(\mathcal{L}^M, \mu')$, there exists a valid matching tree T' that is isomorphic to $\widetilde{T}$ contained in $\overline{U}(\mathcal{L}^M, \mu', S^*)$, where S^* denotes the set of sources in $G(\mathcal{L}^M, \mu)$. If we delete the labels on the vertices in T' (or $\widetilde{T}$) , we get an out-branching (in fact, a matching tree) on $2n + 1$ vertices, denoted by T. Thus, $T \in \mathcal{F}$, and we

Algorithm 4.1: Solves SEOPM.

Input: A set of men and women vertices (M, W), preferences of men $\mathcal{L}^M$, and a partial matching μ'
Let $\mathcal{F} \leftarrow$ {non-isomorphic out-branchings on $2n+1$ vertices}
forall $S \subseteq W$ **do**
  forall *matching tree* $T \in \mathcal{F}$ **do**
    Using Proposition 4 test whether $(\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S), T^{\triangle}, \chi_{sp})$ is a Yes-instance of Col-Sub-Iso.
    if *the algorithm returns a subgraph* T^* **then**
      return T^*
return *"No valid out-branching exists"*

conclude that Algorithm 4.1 will find T^*, a valid triangular matching tree of $\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S^*)$ that is isomorphic to $T^{\triangle}$. Hence, the algorithm will return T^*.

Suppose that Algorithm 4.1 outputs T^*. Then there exists a subset $S \subseteq W$, and an out-branching on $2n+1$ vertices T, such that $\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S)$ contains as subgraph T^* which is isomorphic to the triangular matching tree $T^{\triangle}$. Observe that $\overline{U}^{\triangle}(\mathcal{L}^M, \mu', S)$ has a unique triangle r, r^1, r^2 and thus due to the isomorphism, T^* contains the triangle r, r^1, r^2. This implies that every vertex in T^* is reachable from the root of $\overline{U}(\mathcal{L}^M, \mu', S)$. Since male vertices are only reachable from a female vertex, this means that every male vertex has an in-coming female neighbor. Since, $T^* \setminus \{r^1, r^2\}$ is a valid matching tree of $\overline{U}(\mathcal{L}^M, \mu', S)$, there is exactly one copy of every male vertex and every female vertex has a unique out-neighbor. Thus, if (w_i, m^i_k) is a female to male arc in T^*, then T^* does not contain any other out-going arc from w_i. Thus, the female to male arcs in T^* denote a perfect matching μ. Note that $\mu' \subseteq \mu$ because $U(\mathcal{L}^M, \mu')$ contains a unique out-going arc for every matched woman in μ', hence those arcs must also be part of T^*. Hence, we can conclude that $T^* \setminus \{r^1, r^2\}$ is an out-branching in the graph $\overline{G}\left(\mathcal{L}^M, \mu\right)$. By Proposition 1, this means that $(\mathcal{L}^M, \mu')$ is a Yes-instance of SEOPM. This concludes the proof. ◀

The next lemma gives the running time of Algorithm 4.1.

▶ **Lemma 9.** *Let $(\mathcal{L}^M, \mu')$ be an input to* SEOPM*, where $|M| = n$. Then, Algorithm 4.1 decides whether $(\mathcal{L}^M, \mu')$ is a* Yes*-instance to* SEOPM *in time $2^{\mathcal{O}(n)}$.*

Proof. The running time of the algorithm is upper bounded by the following formula

$|\{S \subseteq W\}| \times |\mathcal{F}| \times$ Time taken by Col-Sub-Iso algorithm

By applying Proposition 5 we upper bound $|\mathcal{F}|$ by $2.956^{2n+1} n^{\mathcal{O}(1)}$. It is a well-known fact that the treewidth of a tree is one, from that it is easy to show that the treewidth of a triangular matching tree is at most 3. (One can first find the tree-decomposition of the tree and then add the two vertices r^1, r^2 to every bag and thus increasing the treewidth by at most two. See [5, Chapter 7] for more details regarding treewidth.) Thus, when we apply Proposition 4, we have a host graph that has at most $n + n^2 + 3$ vertices, and a pattern graph that has size $2n+3$ and treewidth at most 3. Therefore, the running time for using the subroutine for Col-Sub-Iso is $2^{2n+3} n^{\mathcal{O}(1)}$. Multiplying all these values together, gives the overall running time to be $2^n \times 2.956^{2n+1} n^{\mathcal{O}(1)} \times 2^{2n+3} n^{\mathcal{O}(1)} = 2^{\mathcal{O}(n)}$. ◀

Combining Lemmas 8 and 9 we get the following theorem.

▶ **Theorem 10.** *There is an algorithm for* SEOPM *running in time* $2^{\mathcal{O}(n)}$.

Proof. Given an instance $(\mathcal{L}^M, \mu')$ to SEOPM, we first apply Algorithm 4.1. If it returns that "No valid out-branching exists" then we return that $(\mathcal{L}^M, \mu')$ is a No-instance of SEOPM. Else, if the output is T^*, we first obtain T by deleting r^1, r^2 and then using T we obtain a perfect matching $\mu' \subseteq \mu$, by pairing every woman to its unique out-neighbor. Now we invoke **Algorithm Q1** mentioned in [13, Theorem 2] with $(\mathcal{L}^M, \mu)$ and obtain the desired $\mathcal{L}^W$. Correctness and running time follow from Lemmas 8 and 9. This completes the proof. ◀

4.4 A Lower Bound under Exponential Time Hypothesis

In this section we show that Theorem 10 is asymptotically optimal. That is, barring an unlikely scenario occurring in complexity theory, there cannot be a better algorithm for SEOPM. To prove this we will invoke the Exponential Time Hypothesis (ETH), and use the well-known NP-hardness reduction from SAT to SEOPM.

> **Exponential Time Hypothesis (ETH):** Let τ denote the infimum of the set of constants c for which there exists an algorithm solving 3-SAT in time $\mathcal{O}(2^{cn} n^{\mathcal{O}(1)})$. Then it is conjectured that $\tau > 0$.

ETH and its counterpart SETH, introduced by Impagliazzo et al. [11], have been extensively used recently to obtain tight lower bounds for several problems. We use this here to get a lower bound on the running time possible for SEOPM. To this end we will use the following result stated in [5, Theorem 14.4].

▶ **Theorem 11** ([5]). *Unless ETH fails, there exists a constant $c > 0$ such that no algorithm for* 3-SAT *can achieve running time $\mathcal{O}(2^{c(n+m)} n^{\mathcal{O}(1)})$. In particular,* 3-SAT *cannot be solved in time $2^{o(n+m)}$. Here, n and m denote the number of variables and clauses in the input formula to* 3-SAT.

Using Theorem 11 we show the next result.

▶ **Theorem 12.** *Unless ETH fails, there is no algorithm for* SEOPM *running in time* $2^{o(n)}$.

Proof. Let us assume that we can find an algorithm $\mathcal{A}$ that solves SEOPM in time $2^{o(n)}$) where n is the number of men/ women. In [13], Kobayashi and Matsui showed that SEOPM is NP-complete, by giving a reduction from SAT to SEOPM. In particular, given a SAT instance with n variables and m clauses, they reduce it to an instance of SEOPM with $2m + 3n$ men (and women). An easy observation is that in the reduction given by Kobayashi and Matsui [13], we could have started with 3-SAT and reduced it to an instance of SEOPM with $2m + 3n$ men (and women). Now we show how to design an algorithm for 3-SAT running in time $2^{o(n+m)}$ using algorithm $\mathcal{A}$. Given an instance ϕ of 3-SAT, we start by applying the polynomial time reduction given in [13] and obtain an instance of SEOPM with $2m + 3n$ men and $2m + 3n$ women. Now we solve this instance of SEOPM using algorithm $\mathcal{A}$ in time $2^{o(m+n)}$. Using the solution to an instance of SEOPM we decide in polynomial time whether ϕ is satisfiable or not. Thus, we have given an algorithm for 3-SAT running in time $2^{o(n+m)}$, contradicting Theorem 11. This concludes the proof. ◀

5 Concluding thoughts

In this paper we designed an exact algorithm for Stable Extension of Partial Matching running in time $2^{\mathcal{O}(n)}$. We complemented this result by showing that unless ETH fails the running time bound is asymptotically optimal. There are several problems in the stable matching model that are NP-complete and have been studied from the perspective of approximation algorithms. However, there is almost no study about these problems either from the view point of moderately exponential time algorithms or parameterized complexity. The area needs a thorough study in these algorithmic paradigms and is waiting to explode.

References

1 Omid Amini, Fedor V. Fomin, and Saket Saurabh. Counting subgraphs via homomorphisms. *SIAM J. Discrete Math.*, 26(2):695–717, 2012.

2 Terry Beyer and Sandra Mitchell Hedetniemi. Constant time generation of rooted trees. *SIAM Journal on Computing*, 9(4):706–712, 1980.

3 Robert Bredereck, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, and Gerhard J. Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014.

4 A. Cseh and D. F. Manlove. Stable marriage and roommates problems with restricted edges: complexity and approximability. In *Proceedings of SAGT'15*, volume 9347 of *LNCS*, pages 15–26, 2015.

5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

6 Fedor V Fomin and Dieter Kratsch. *Exact exponential algorithms*. Springer Science & Business Media, 2010.

7 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative sets of product families. In *ESAs*, volume 8737 of *Lecture Notes in Computer Science*, pages 443–454. Springer, 2014.

8 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, Saket Saurabh, and B. V. Raghavendra Rao. Faster algorithms for finding and counting subgraphs. *J. Comput. Syst. Sci.*, 78(3):698–706, 2012.

9 David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

10 D. Gusfield and R. W. Irving. *The Stable Marriage Problem-Structure and Algorithm*. The MIT Press, 1989.

11 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

12 H. Kobayashi and T. Matsui. Successful manipulation in stable marriage model with complete preference. *IEICE Trans. on Inf. Syst.*, E92-D(2):116–119, 2009.

13 H. Kobayashi and T. Matsui. Cheating strategies for the gale-shapley algorithm with complete preference lists. *Algorithmica*, 58:151–169, 2010.

14 D. F. Manlove. *Algorithmics of matching under preferences*, volume 2 of *Theoretical Computer Science*. World Scientific, 2013.

15 Dániel Marx and Ildikó Schlotter. Parameterized complexity and local search approaches for the stable marriage problem with ties. *Algorithmica*, 58(1):170–187, 2010.

16 Dániel Marx and Ildikó Schlotter. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optimization*, 8(1):25–40, 2011.

17 Richard Otter. The number of trees. *Annals of Mathematics*, pages 583–599, 1948.

18 A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game Theoretic Modeling and Analysis*. Cambridge Univ. Press, 1990.