Complexity Study for the Robust Stable Marriage Problem

Begum Genc, Mohamed Siala, Gilles Simonin, Barry O'Sullivan

Complexity Study for the Robust Stable Marriage Problem ☆

Begum Genc[a,*], Mohamed Siala[b], Gilles Simonin[c], Barry O'Sullivan[a]

[a]*Insight Centre for Data Analytics, School of Computer Science,*
University College Cork, Ireland
[b]*LAAS-CNRS, Universit de Toulouse, CNRS, INSA, Toulouse, France*
[c]*IMT Atlantique, LS2N, CNRS, UBL, 44307 Nantes, France*

Abstract

The Robust Stable Marriage problem (RSM) is a variant of the classic *Stable Marriage* problem in which the robustness of a given stable matching is measured by the number of modifications required to find an alternative stable matching should some pairings break due to an unforeseen event. We focus on the complexity of finding an (a, b)-supermatch. An (a, b)-supermatch is defined as a stable matching in which if any a (non-fixed) men/women break up it is possible to find another stable matching by changing the partners of those a men/women and the partners of at most b others. We first discuss a model based on independent sets for finding $(1, 1)$-supermatches. Secondly, in order to show that deciding whether or not there exists a $(1, b)$-supermatch is $\mathcal{NP}$-complete, we first introduce a SAT formulation for which the decision problem is $\mathcal{NP}$-complete by using Schaefer's Dichotomy Theorem. We then show the equivalence between this SAT formulation and finding a $(1, 1)$-supermatch on a specific family of instances. We also focus on studying the threshold between the cases in P and $\mathcal{NP}$-complete for this problem.

Keywords: Stable Matching, Robust Stable Marriage, Robustness, Schaefer's Dichotomy, NP-completeness.

1. Introduction

Matching under preferences is a multidisciplinary family of problems, mostly studied by researchers in the fields of economics and computer science. This family of problems has many applications in real-world applications such as house allocation, college admission, kidney exchange, etc. We refer the reader to [20] for a comprehensive review of the subject. The Stable Marriage problem (SM), first introduced by Gale and Shapley in [6], is the central problem in this context. In the SM, the purpose is to match men and women while respecting their preferences. To be more precise, every person gives his/her (strict) preferences over persons of the opposite sex. A solution to the SM is a matching M between men and women such that: (1) every person is matched to another person from the opposite sex; and (2) no pair $\langle man, woman \rangle$ prefers each other to their assigned partners in M. Any such matching is called a stable matching.

In this paper we study the Stable Marriage problem from a *"robust optimization"* perspective. Robust optimization is a broad area that dates back to the 1970s [24]. In general, in a robust optimization model, uncertain parameters that are derived from noisy, incomplete, or erroneous data are modelled as random variables with discrete distributions [27].

We find in the literature a variety of definitions of robustness depending on the context, for example in scheduling and timetabling [22], constraint programming [3], machine learning [26], and economics [25]. Some researchers define a solution to be robust if it remains a solution after an unexpected occurrence of an event. Others, however, refer to a solution as robust if an alternative solution can be found at a small cost (when dealing with an unexpected event).

Robustness in matching problems is a relatively new and important topic. The first appearance of robustness dates back to 2011 within the context of matching markets [18]. Kojima's robustness notion is motivated by students misreporting their preferences to manipulate their assignment to schools. Afacan, on the other hand, extended the concept of robust stability proposed by Kojima to *group robust stability*. In this context, a group of students misrepresent their choices for manipulation [1].

Some researchers have focused on finding robust solutions to stable matchings when uncertainty in the preferences is present. Drummond and Boutilier

proposed to use minimax regret as a measure of robustness of stable matchings for the stable marriage problem [5]. The use of minimax regret provides robustness by minimizing the worst-case loss. The work is motivated by situations in which incomplete information is provided by the agents.

A recent study from Menon and Larson focuses on finding a matching that minimizes the number of expected blocking pairs, and refer to a good solution as a robust one [21]. They mostly focus on providing algorithms to find such matchings. Aziz et al. considered different models to study this problem and mostly focused on the complexity part of the problem [2]. They define *stability probability* as the probability of a matching being stable. They then model a number of problems related to stability probability using the uncertainty and provide rich complexity results.

One of the most recent robustness notions is proposed by Jacobovic for the stable marriage problem. They use a probability model and a social cost function to measure robustness [17]. The work is motivated by settings in which some agents leave the stable matching after it has been constructed. Mai and Vazirani also work on a robust version of the stable marriage problem [19]. Their robustness notion is defined over the errors in the input.

In this paper, we focus on the robustness notion proposed by Genc et al. [9, 7] related to the concept of an (a, b)-supermatch. Informally, an (a, b)-*supermatch* is a stable matching such that if any (non-fixed) a men break-up, it is possible to find another stable matching by changing the partners of those a men and at most b others. Note that, the term "men" can be replaced by "women" in the definition of (a, b)-supermatches, but throughout this work, we use "men" to be consistent with convention in the literature.

The (a, b)-supermatch concept is inspired by (a, b)-supermodels in Boolean satisfiability [10] and (a, b)-super solutions in constraint satisfaction [15, 14]. An (a, b)-supermodel is a model such that if the values taken by the variables in a set of size at most a are reassigned (breakage), another model can be obtained by modifying the values of the variables in an additional disjoint set of size at most b (repair) variables [10]. Similarly, an (a, b)-super solution is a solution in which if any a variables break, the solution can be repaired by providing repair by changing a maximum of b other variables [13]. Finding an (a, b)-supermodel and finding an (a, b)-super solution are known to be $\mathcal{NP}$-hard.

In our earlier work we defined a polynomial-time procedure to verify if a given stable matching to a stable marriage instance is a $(1, b)$-supermatch. We have also proposed three approaches based on a constraint programming

3

model, an iterated local search procedure, and a genetic algorithm procedure, to solve the optimization problem of finding a $(1, b)$-supermatch with the minimum b. All the three models rely on the polynomial time verification procedure. However, the complexity of finding (a, b)-supermatches was left as an open problem.

In this paper we study the complexity of finding an (a, b)-supermatch. We first show that the decision problem for finding a $(1, 1)$-supermatch on a restricted family of instances is $\mathcal{NP}$-complete using Schaefer's dichotomy theorem. Then, we generalize the result to the $(1, b)$ case. However, the problem of finding (a, b)-supermatches, while being $\mathcal{NP}$-hard, is an open question whether it belongs to the class $\mathcal{NP}$ or not. Next, we present a family of instances for which a $(1, 1)$-supermatch can be found in polynomial time. We also prove that $(2, 0)$-supermatches do not exist. Finally, we discuss the existence of $(a, 0)$-supermatches.

The remainder of the paper is organised as follows. In Section 2 the notation and the basics of the stable marriage problem are introduced. We also present some important theorems in the same section. We present a representation for identifying $(1, 1)$-supermatches using maximal independent sets in Section 3. In Section 4.1 we describe a restricted family of stable marriage instances. Following this definition, we define a specific SAT formulation based on the properties of the restricted family in Section 4.2. We conclude the $\mathcal{NP}$-completeness of finding an $(1, 1)$-supermatch in Section 4.3 by showing the equivalence to the restricted family and the SAT formulation. Finally, in Section 5, we discuss some special cases for the general problem of Robust Stable Marriage.

2. Notation and Background

2.1. The Stable Marriage Problem

An instance of the *Stable Marriage problem (with incomplete lists)* takes as input a set of men $U = \{m_1, m_2, \ldots, m_{n_1}\}$ and a set of women $W = \{w_1, w_2, \ldots, w_{n_2}\}$ where each person has an ordinal preference list over members of the opposite sex. For the sake of simplicity we assume in the remainder of this paper that $n_1 = n_2$. A *pair* $\langle m_i, w_j \rangle$ is acceptable if w_j (respectively m_i) appears in the preference list of m_i (respectively w_j). A matching is a set of acceptable pairs where each man (respectively woman) appears at most once in any pair of M. If $\langle m_i, w_j \rangle \in M$, we say that w_j (respectively m_i)

4

is the partner of m_i (respectively w_j) and then we denote $M(m_i) = w_j$ and $M(w_j) = m_i$.

A pair $\langle m_i, w_j \rangle$ (sometimes denoted as $\langle i, j \rangle$) is said to be *blocking* a matching M if m_i is unassigned or prefers w_j to $M(m_i)$ and w_j is unassigned or prefers m_i to $M(w_j)$. A *matching* M is called *stable* if there exists no blocking pair for M. A *pair* $\langle m_i, w_j \rangle$ is said to be *stable* if it appears in a stable matching. The stable matching in which each man is matched to their most preferred stable partner is called the man-optimal matching and denoted by M_0.

A pair $\langle m_i, w_j \rangle$ is called *fixed* if $\langle m_i, w_j \rangle$ appears in every stable matching. In this case, the man m_i and woman w_j are called fixed. In the remainder of this paper we use n to denote the number of non-fixed men and $\mathcal{I}$ to denote an instance of a stable marriage problem. We measure the distance between two stable matchings M_i, M_j by the number of men that have different partners in M_i and M_j, denoted by $d(M_i, M_j)$.

Formally, a stable matching M is said to be (a, b)-*supermatch* if for any set $\Psi \subseteq M$ of a stable pairs that are not fixed, there exists a stable matching M' such that $M' \cap \Psi = \emptyset$ and $d(M, M') \leq a + b$, from [7]. It is important to note here that the Rural Hospitals Theorem (see Theorem 1.4.2 in [12]) states that the same set of agents are assigned in all stable matchings. Therefore, if any a men who are matched in M lose their partners, we cannot expect to find another stable matching without those a men becoming matched again.

Definition 1 (π_{ab}). *Decision problem for (a, b)-supermatch.*
INPUT*: $a, b \in \mathbb{N}$, and a Stable Marriage instance $\mathcal{I}$.*
QUESTION*: Is there an (a, b)-supermatch for $\mathcal{I}$?*

Let M be a stable matching and $\Psi \subseteq M$ be a set of non-fixed pairs. A *repair matching* for the breakage of Ψ in M is a stable matching M' that minimizes the value of $d(M, M^*)$ taken over every other stable matching M^* such that $M^* \cap \Psi = \emptyset$. The *repair cost* in this context is the value $d(M, M') - |\Psi|$.

A **rotation** $\rho = (\langle m_{k_0}, w_{k_0} \rangle, \langle m_{k_1}, w_{k_1} \rangle, \ldots, \langle m_{k_{l-1}}, w_{k_{l-1}} \rangle)$, where $l \in \mathbb{N}^*$, is an ordered list of pairs in a stable matching M such that changing the partner of each man m_{k_i} to the partner of the next man $m_{k_{i+1}}$ (the operation $+1$ is modulo l) in the list ρ leads to a stable matching denoted by M/ρ. The latter is said to be obtained after eliminating ρ from M. In this case, we say that $\langle m_{l_i}, w_{l_i} \rangle$ is *eliminated* by ρ, whereas $\langle m_{l_i}, w_{l_{i+1}} \rangle$ is *produced* by ρ, and that ρ is *exposed* in M. If a pair $\langle m_i, w_j \rangle$ appears in a rotation ρ we

denote it by $\langle m_i, w_j \rangle \in \rho$. Additionally, if a man m_i appears at least in one of the pairs in the rotation ρ, we say that m_i is *involved* in ρ or ρ *contains* man m_i.

There exists a partial order for rotations. A rotation ρ' is said to precede another rotation ρ (denoted by $\rho' \prec\prec \rho$) if ρ' is eliminated in every sequence of eliminations that starts at M_0 and ends at a stable matching in which ρ is exposed [12]. In this case, we say ρ' is the *predecessor* of ρ. Similarly, ρ is the *successor* of ρ'. Additionally, given two rotations ρ and ρ', we say that ρ' is an *immediate predecessor* of ρ if $\rho' \prec\prec \rho$ and there is no rotation ρ'' such that $\rho' \prec\prec \rho'' \prec\prec \rho$. Similarly ρ is an *immediate successor* of ρ' if ρ' is an immediate predecessor of ρ. Immediate predecessors of a rotation ρ in a rotation poset are denoted by $N^-(\rho)$ and immediate successors are denoted by $N^+(\rho)$. We also denote by $X(R)$ the set of men involved in a set of rotations R.

The structure that represents all rotations and their partial order is a directed graph called ***rotation poset*** denoted by $\Pi = (\mathcal{V}, E)$. Each rotation corresponds to a node in $\mathcal{V}$ and there exists an edge from ρ' to ρ if ρ' precedes ρ. There are two different edge types in a rotation poset: ***Type 1*** and ***Type 2***. Suppose that a pair $\langle m_i, w_j \rangle$ is in a rotation ρ. If ρ' is the unique rotation that moves m_i to w_j then $(\rho', \rho) \in E$ and ρ' is called a Type 1 predecessor of ρ. Suppose now that a pair $\langle m_i, w_j \rangle$ is not involved in any rotations. If ρ moves m_i below w_j, and $\rho' \neq \rho$ is the unique rotation that moves w_j above m_i, then $(\rho', \rho) \in E$ and ρ' is called a Type 2 predecessor of ρ [12]. A node that has no outgoing edges is called a *sink node* and a node that has no incoming edges is called *source node*.

A ***closed subset*** S is a set of rotations such that for any rotation ρ in S, if there exists a rotation ρ' that precedes ρ then ρ' is also in S. There is a one-to-one correspondence between closed subsets and stable matchings [12]. Let $L(S)$ be the set of rotations that are the sink nodes of the graph induced by the rotations in S. Similarly, let $N(S)$ be the set of the rotations that are not in S, but all of their predecessors are in S. This can be illustrated as having a cut in the graph Π, where the cut divides Π into two subgraphs, namely Π_1 and Π_2. If there are any comparable nodes between Π_1 and Π_2, Π_1 is the part that contains the preceding rotations. Eventually, Π_1 corresponds to the closed subset S, $L(S)$ corresponds to the sink nodes of Π_1 and $N(S)$ corresponds to the source nodes of Π_2.

An important remark is that there is a one-to-one correspondence between the stable matchings in $\mathcal{I}$ and the sets of incomparable rotations in $\mathcal{V}$ as

6

shown in Proposition 1. A closed subset is defined by adding all predecessors of each node in the subset to the subset. Equivalently, if all rotations that precede some other rotations in S are removed from S, the resulting set corresponds to a set of incomparable nodes, namely $\mathrm{L}(S)$.

Proposition 1. *There is a one-to-one correspondence between the incomparable rotations and the stable matchings of the underlying instance.*

Proof. $\Rightarrow$ Let V_I denote a set of incomparable rotations. By adding all the predecessors of the rotations in set V_I, we obtain a closed subset. The latter defines a stable matching.
$\Leftarrow$ Let M be a stable matching and let S be its closed subset. The set of sink nodes of S corresponds (by definition) to a set of incomparable rotations. $\square$

Irving and Leather previously showed that the number of stable matchings in the Stable Marriage problem grows exponentially with respect to the number of men/women [16]. However, the rotation poset provides a compact and powerful representation for all stable matchings. The number of rotations in a rotation poset Π is bounded by $n(n-1)/2$ and the number of edges is bounded by n^2.

Let us illustrate these terms on a sample SM instance specified by the preference lists of 7 men/women in Table 1 given by Genc et. al [7]. For the sake of clarity, each man m_i is denoted with i and each woman w_j with j in the preference lists. Table 2 represents the rotation poset and all the rotations associated with this example.

m_1	1 7 6 3 5 2 4
m_2	7 2 5 6 1 3 4
m_3	7 1 4 2 6 5 3
m_4	4 3 1 2 5 7 6
m_5	2 3 1 4 5 6 7
m_6	7 2 1 4 6 5 3
m_7	3 6 1 7 5 4 2

w_1	3 2 7 5 6 4 1
w_2	1 5 4 6 3 7 2
w_3	3 6 1 5 4 2 7
w_4	7 2 3 4 5 1 6
w_5	5 7 1 6 4 2 3
w_6	4 2 3 7 6 5 1
w_7	5 7 3 2 4 1 6

Table 1: Preference lists for men (left) and women (right) for a sample instance of size 7.
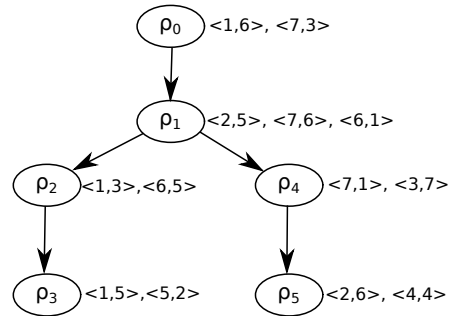
Table 2: Rotation poset of the instance given in Table 1.

7

Table 3: The list of all stable matchings for the instance given in Table 1.

Stable Matching	Pairs
M_0	$\{(1,6),(2,5),(3,7),(4,4),(5,2),(6,1),(7,3)\}$
$M_1 = M_0/\rho_0$	$\{(1,3),(2,5),(3,7),(4,4),(5,2),(6,1),(7,6)\}$
$M_2 = M_1/\rho_1$	$\{(1,3),(2,6),(3,7),(4,4),(5,2),(6,5),(7,1)\}$
$M_3 = M_2/\rho_4$	$\{(1,3),(2,6),(3,1),(4,4),(5,2),(6,5),(7,7)\}$
$M_4 = M_3/\rho_5$	$\{(1,3),(2,4),(3,1),(4,6),(5,2),(6,5),(7,7)\}$
$M_5 = M_2/\rho_2$	$\{(1,5),(2,6),(3,7),(4,4),(5,2),(6,3),(7,1)\}$
$M_6 = M_5/\rho_4 = M_3/\rho_2$	$\{(1,5),(2,6),(3,1),(4,4),(5,2),(6,3),(7,7)\}$
$M_7 = M_4/\rho_2 = M_6/\rho_5$	$\{(1,5),(2,4),(3,1),(4,6),(5,2),(6,3),(7,7)\}$
$M_8 = M_5/\rho_3$	$\{(1,2),(2,6),(3,7),(4,4),(5,5),(6,3),(7,1)\}$
$M_9 = M_6/\rho_3 = M_8/\rho_4$	$\{(1,2),(2,6),(3,1),(4,4),(5,5),(6,3),(7,7)\}$
$M_{10} = M_7/\rho_3 = M_9/\rho_5$	$\{(1,2),(2,4),(3,1),(4,6),(5,5),(6,3),(7,7)\}$

All the stable matchings of the instance given in Table 1 are listed in Table 3. For the sake of the example, the stable matching M_2 has the corresponding closed subset S_2, which is composed of the rotations $\{\rho_0, \rho_1\}$, and it is obtained by exposing the rotation ρ_1 in M_1, also denoted by $M_2 = M_1/\rho_1 = \{(1,3),(2,6),(3,7),(4,4),(5,2),(6,5),(7,1)\}$. For M_2, the sink and neighbor nodes can be identified as $\mathrm{L}(S_2) = \{\rho_1\}$ and $\mathrm{N}(S_2) = \{\rho_2, \rho_4\}$.

We now demonstrate how to find b to show that M_2 is a $(1,b)$-supermatch. One can simply compute the distance from M_2 to all other stable matchings listed in Table 3 to find the closest stable matching in case of the breakage of each pair. However, we use the polynomial time procedure to find the value of b proposed by Genc et al, but we do not discuss the details of the proposed procedure in this paper [7]. Assume that S' denotes the closed subset of the closest stable matching to M_2 after breakage of the given pair.

- $\langle m_1, w_3 \rangle$: $S' = \{\rho_0, \rho_1, \rho_2\}$, corresponds to M_5, $d(M_2, M_5) = 2$

- $\langle m_2, w_6 \rangle$: $S' = \{\rho_0\}$, corresponds to M_1, $d(M_2, M_1) = 3$

- $\langle m_3, w_7 \rangle$: $S' = \{\rho_0, \rho_1, \rho_4\}$, corresponds to M_3, $d(M_2, M_3) = 2$

- $\langle m_4, w_4 \rangle$: $S' = \{\rho_0, \rho_1, \rho_4, \rho_5\}$, corresponds to M_4, $d(M_2, M_4) = 4$

- $\langle m_5, w_2 \rangle$: $S' = \{\rho_0, \rho_1, \rho_2, \rho_3\}$, corresponds to M_8, $d(M_2, M_8) = 3$

- $\langle m_6, w_5 \rangle$: $S' = \{\rho_0, \rho_1, \rho_2\}$, corresponds to M_5, $d(M_2, M_5) = 2$

- $\langle m_7, w_1 \rangle$: $S' = \{\rho_0, \rho_1, \rho_4\}$, corresponds to M_3, $d(M_2, M_3) = 2$

Considering that $a = 1$, the b values are computed by subtracting 1 from all the computed distance values. Then, one can infer that if any of the men from the following set $\{m_1, m_3, m_6, m_7\}$ wants to break up, an alternative matching at a repair cost of 1 can be found. For men m_2, m_5 the cost is 2 and for m_4 the cost is 3. Hence, M_2 is a $(1, 3)$-supermatch, meaning if any of the men break their matches in M_2, an alternative matching can be found by changing at most 3 other men's partners. Additionally, the reader can verify that stable matching M_6 with the corresponding closed subset $S_6 = \{\rho_0, \rho_1, \rho_2, \rho_4\}$ is a $(1, 1)$-supermatch.

In order to check if a stable matching is a $(2, b)$-supermatch, currently there does not exist a polynomial-time procedure to check this. However, one can compare the given stable matching with all the other ones that contain different partners for the men for all possible combinations of size 2. Any such stable matching that minimizes the distance with the given one is computed as the repair matching for the break-up of those 2 men.

2.2. Schaefer's Dichotomy Theorem for Boolean Satisfiability

Schaefer's dichotomy theorem was presented in [23]. In this section, we use the same terminology and notation as in [4]. A **literal** is a Boolean variable or its negation. A **clause** is a disjunction of literals. If x is a Boolean variable, then the literal x is called *positive* and the literal $\neg x$ is called *negative*. We shall use the term *formula* to say a Boolean formula given in a conjunctive normal form (**CNF**) as a finite set of clauses.

A formula is called **Horn** (respectively **dual-Horn**) if every clause in this formula contains at most one positive (respectively negative) literal. A *linear equation over the 2-element field* is an expression of the form $x_1 \oplus x_2 \ldots \oplus x_k = \delta$ where $\oplus$ is the sum modulo 2 operator and δ is 0 or 1. An **affine formula** is a conjunction of linear equations over the 2-element field.

An *assignment* is a mapping from (Boolean) variables to $\{true, false\}$. An assignment A is said to satisfy a clause C if and only if there exists a variable x such that C contains x and the assignment of x by A is *true* or C contains $\neg x$ and the assignment of x by A is *false*.

A **Boolean Constraint of arity** k is a function $\phi : \{true, false\}^k \to \{true, false\}$. Let $(x_1, \ldots x_k)$ be a sequence of Boolean variables and ϕ be

9

a Boolean constraint of arity k. The pair $\langle \phi, (x_1, \ldots x_k) \rangle$ is called a *constraint application*. An assignment A to $(x_1, \ldots x_k)$ satisfies $\langle \phi, (x_1, \ldots x_k) \rangle$ if ϕ evaluates to *true* on the truth values assigned by A. Let Φ be a set of constraint applications, and A be an assignment to all variables occurring in Φ. A is said to be a *satisfying assignment* of Φ if A satisfies every constraint application in Φ.

Let $\mathcal{C}$ be a set of Boolean constraints. $\boldsymbol{SAT(\mathcal{C})}$ is defined as the following decision problem: given a finite set Φ of constraints applications from $\mathcal{C}$, is there a satisfying assignment for Φ?

Theorem 1. *Dichotomy Theorem for Satisfiability by [4, 23]. Let $\mathcal{C}$ be a set of Boolean constraints. If $\mathcal{C}$ satisfies at least one of the conditions (a)-(f) below, then $SAT(\mathcal{C})$ is in $\mathcal{P}$. Otherwise, $SAT(\mathcal{C})$ is $\mathcal{NP}$-complete.*

  a) *Every constraint in $\mathcal{C}$ evaluates to true if all assignments are true.*

  b) *Every constraint in $\mathcal{C}$ evaluates to true if all assignments are false.*

  c) *Every constraint in $\mathcal{C}$ can be expressed as a Horn formula.*

  d) *Every constraint in $\mathcal{C}$ can be expressed as a dual-Horn formula.*

  e) *Every constraint in $\mathcal{C}$ can be expressed as affine formula.*

  f) *Every constraint in $\mathcal{C}$ can be expressed as a 2-CNF formula.*

3. A Representation for Finding $(1, 1)$-Supermatches

We describe a representation for the problem of finding $(1, 1)$-supermatches. Our representation is based on finding independent sets with additional constraints. Independent sets have been used previously to find stable matchings in the Stable Roommates problem (SR) [12]. For SR, the authors show that each maximal independent set on an undirected graph derived from its rotation poset is in one-to-one correspondence with the stable matchings of the underlying instance. However, for the Robust Stable Marriage problem, we show that the additional constraints change the structure of the problem, and therefore the approach is different.

An *independent set* in an undirected graph G is defined as a set of nodes such that no two nodes from the set share an edge in G. Consider a Stable

Marriage instance $\mathcal{I}$ and let $\Pi = (\mathcal{V}, E)$ be its rotation poset. We use $\Pi_u = (\mathcal{V}, E_u)$ to denote the undirected representation of Π with transitivity. It is important to observe that, every independent set defined in Π_u consists of incomparable rotations. Since Π_u contains transitive edges meaning if a rotation ρ' is preceding another rotation ρ, there exists an edge $(\rho', \rho) \in E_u$, and therefore the rotations ρ' and ρ together cannot be members of any independent sets of Π_u. Using Proposition 1 and the observation, it is correct to state that there exists a one-to-one correspondence between the independent sets of Π_u and the stable matchings.

Let us also introduce a function $R^*(V)$, which takes a set of rotations denoted by V as input, and returns only the rotations that are of size 2 in this set. Also recall that $X(V)$ denotes the set of men involved in at least one of the rotations in V.

Proposition 2. *Let M be a stable matching and S be its closed subset. M is a $(1, 1)$-supermatch if and only if every non-fixed men of the instance are included in a neighbor or sink rotation of S of size 2, i.e. $|X(R^*(\mathrm{L}(S) \cup \mathrm{N}(S)))| = n$, where n is the number of non-fixed men.*

Proof. $\Rightarrow$ Suppose for contradiction that $|X(R^*(\mathrm{L}(S) \cup \mathrm{N}(S)))| \neq n$. Take a (non-fixed) man m that is not in $X(R^*(\mathrm{L}(S) \cup \mathrm{N}(S)))$. In order to repair M when m breaks up with his partner, one needs to remove or add a rotation of size at least 3 or a set of rotations where at least 3 other men are modified. Therefore M cannot be $(1, 1)$-supermatch (contradiction).
$\Leftarrow$ If $|X(R^*(\mathrm{L}(S) \cup \mathrm{N}(S)))| = n$, then there exists a repair for the breakage of every man by either adding or removing a single rotation of size 2 to/from S. Therefore, M corresponds to a $(1, 1)$-supermatch. $\square$

Given a set of nodes I in $\mathcal{V}$, let P_I be the set of nodes that corresponds to rotations that precede a rotation from I. We use $\Pi^I_{sub} = (\mathcal{V}^I_{sub}, E_{sub})$ to denote the subgraph induced by Π where the set of nodes is $\mathcal{V}^I_{sub} = \mathcal{V} \setminus (P_I \cup I)$. We also define a function $source(G)$ that takes a directed graph G as input and returns the nodes in G that do not have any predecessors. The following theorem gives an formulation of the problem of finding $(1, 1)$-supermatches. It is immediate from Propositions 1 and 2.

Theorem 2. *There exists a one-to-one correspondence between $(1, 1)$-supermatches and the set of the independent sets $\mathcal{S}$ that satisfies $I \in \mathcal{S}$ if and only if $|X(R^*(I \cup source(\Pi^I_{sub})))| = n$.*

Proof. By construction. □

For illustration, consider the case of the Stable Marriage instance of Table 1 and its rotation poset Π given in Table 2. The version of Π, where the transitive edges are included Π_u is given in Figure 1.
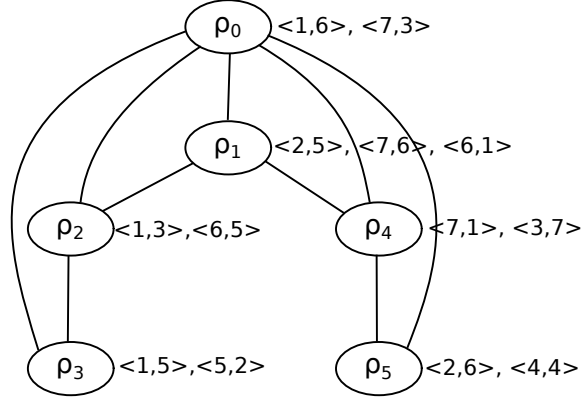

Figure 1: Undirected graph representation with transitive edges included of the rotation poset given in Figure 2.

A sample independent set on Π_u can be given as $I_i = \{\rho_2, \rho_4\}$. The rotations of size 2 are $R^*(I_i) = I_i$ as all the rotations in I_i are of size 2. The subgraph $\Pi_{sub}^{I_i}$ contains only the vertices ρ_3 and ρ_5 and an empty set of edges. The source nodes of $\Pi_{sub}^{I_i}$ are therefore $\{\rho_3, \rho_5\}$ and they are all of size 2. The set of men $X(R^*(I_i \cup source(\Pi_{sub}^{I_i}))) = \{1, 2, 3, 4, 5, 6, 7\}$ and its size is 7, equal to the number of non-fixed men n in the problem. Therefore, according to Theorem 2, the independent set I_i corresponds to a $(1, 1)$-supermatch.

The closed subset that corresponds to I_i is obtained by adding all the preceding rotations P_{I_i} of the rotations in I_i to I_i, yielding in $S_i = \{\rho_0, \rho_1, \rho_2, \rho_4\}$. The closed subset S_i, and the corresponding stable matching M_i can be verified from Table 3 as M_6. It can also be verified that M_6 is a $(1, 1)$-supermatch.

A second illustration on the same graph is with the independent set $I_j = \{\rho_0\}$, corresponding to stable matching M_1 in Table 3. The rotation set $R^*(I_j) = I_j$, and $\Pi_{sub}^{I_j}$ consists of the rotations $\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_5\}$. Since ρ_1 is the only source node (with size 3), we have, $X(R^*(I_j \cup (source(\Pi_{sub}^{I_j}))) = \{1, 7\}$ and the size of this set is $2 < 7$. Therefore, the stable matching corresponding to the closed subset $\{\rho_0\}$ is not a $(1, 1)$-supermatch.

4. Intractability

In this section we show that deciding if there exists a $(1,1)$-supermatch for a given Stable Marriage instance is $\mathcal{NP}$-complete. We begin by defining a specific family of SM instances. We then introduce a specific SAT problem with many conditions and rules. This SAT problem actually is a SAT formulation of the problem of finding a $(1,1)$-supermatch on this restricted family of SM instances. After defining the SAT formulation, we show using the Schaefer's Dichotomy Theorem that the formulation is $\mathcal{NP}$-complete. Subsequently, we state our complexity result by showing the equivalence between the restricted SAT formulation and the problem of deciding if there exists a $(1,1)$-supermatch for the restricted family of instances. We also provide examples to help the reader to understand the steps in the rest of the section.

4.1. A Specific Problem Family

In this section, we describe a restricted, specific family F of Stable Marriage instances over properties on its generic rotation poset $\Pi_F = (\mathcal{V}_F, E_F)$.

Property 1 Each rotation $\rho_i \in \mathcal{V}_F$, contains exactly 2 pairs $\rho_i = (\langle m_{i1}, w_{i1} \rangle, \langle m_{i2}, w_{i2} \rangle)$.

Property 2 Each rotation $\rho_i \in \mathcal{V}_F$, has at most 2 immediate predecessors and at most 2 immediate successors.

Property 3 Each edge $e_i \in E_F$, is a Type 1 edge.

Property 4 For each man $m_i, i \in [1, n]$, m_i is involved in at least 2 rotations.

Figure 2 illustrates these properties. We denote each pair $\langle m_i, w_j \rangle$ by $\langle i, j \rangle$ to make it easier to read. The different cases in the figure emphasize Property 2, where a rotation has exactly: one predecessor and one successor (A,B); one predecessor and two successors, which is also similar to having one successor and two predecessors (C); two predecessors and two successors (D). It can be verified that Properties 1 and 3 are satisfied in all of the cases. However, Property 4 is not satisfied as this is not a complete example. Observe that, the ordering of the pairs is not important as there exist only two pairs in each rotation.

We would like to emphasize the difference between the cases A and B in Figure 2. Due to Property 3, any two rotations that have an edge between
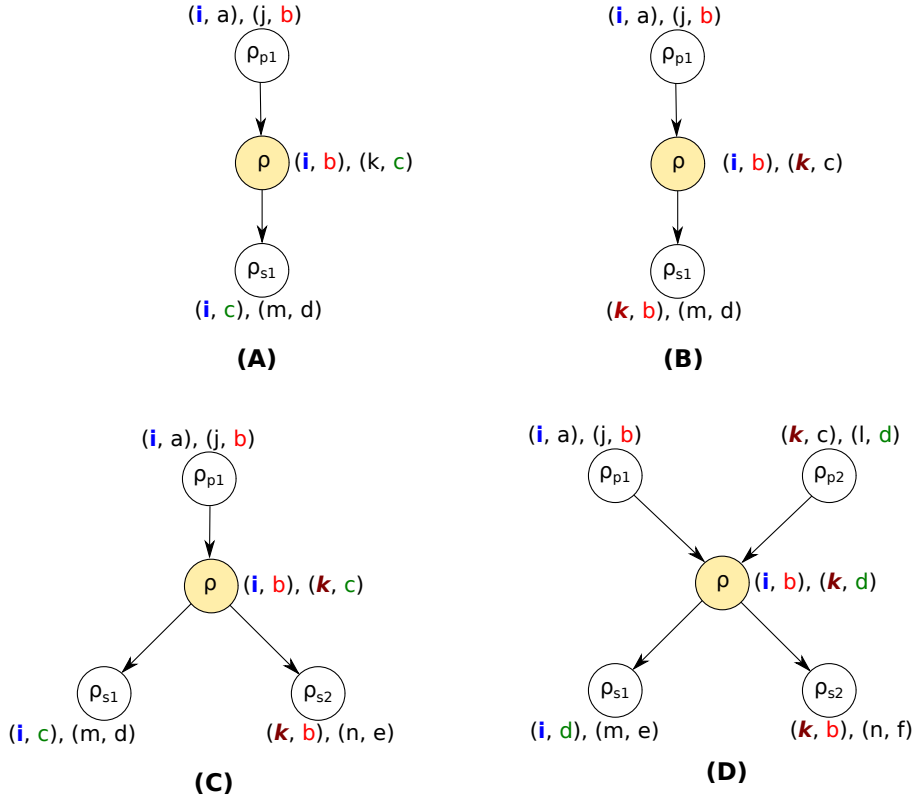
Figure 2: Illustrations of different cases for the men and women included in the rotations in Π of SM instances in F.

them contain a man and a woman in common. When the case is generalized to three rotations $\rho_{p1} - \rho - \rho_{s1}$, it should be noted that those three rotations either contain the same man m_i but contain different women, w_b between ρ_{p1} and ρ, but w_c between ρ and ρ_{s1}, as in Case A, or the same woman in all three rotations but different men m_i between ρ_{p1} and ρ, but m_k between ρ and ρ_{s1}, as in the Case B.

Lemma 1 is a characterization for all the SM instances and can easily be observed on the family F.

Lemma 1. *For every two different paths P_1 and P_2 defined on Π_F, where both start at rotation ρ_s, end at ρ_t, and the pair $\langle m_e, w_f \rangle \in \rho_s$, if all rotations on P_1 (respectively P_2) contain m_e, at least one of the rotations on P_2 (respectively P_1) does not contain w_f.*

Proof. Suppose for a contradiction that m_e is involved in all rotations on P_1,

14

Table 4: An SM instance from family F of 6 men and 6 women.

m_1	1 2 6 3 4 5	w_1	3 2 1 4 5 6
m_2	2 1 4 3 5 6	w_2	5 6 1 2 3 4
m_3	3 4 1 2 5 6	w_3	6 5 4 3 1 2
m_4	4 3 5 1 2 6	w_4	2 3 4 1 5 6
m_5	5 3 2 1 4 6	w_5	4 5 1 2 3 6
m_6	6 2 3 1 4 5	w_6	1 6 2 3 4 5

and w_f is involved in all rotations on P_2. This scenario is likely to occur as Property 3 indicates that all edges are of Type 1, which is also easy to observe in Figure 2, where any two rotations connected by an edge in the rotation poset always contain a man and a woman in common. In this case, because of the supposition, namely that m_e and w_f are carried on to the rotation ρ_t, the pair $\langle m_e, w_f \rangle$ is reproduced. In other words, exposing ρ_t on a stable matching produces the pair $\langle m_e, w_f \rangle$. However, this pair is already eliminated by ρ_s, meaning this couple is already produced. The supposition contradicts the fact that exposing rotations on stable matchings causes men to be matched with their less preferred partners, and if a couple is eliminated once, it cannot be produced again. $\qquad \square$

In Table 4 we provide a sample SM instance $\mathcal{I}$ from family F. The instance contains 6 men and 6 women.

Figure 3 illustrates the rotation poset of this instance. It can be easily verified that Properties 1, 2, 3 and 4 apply to the rotation poset.
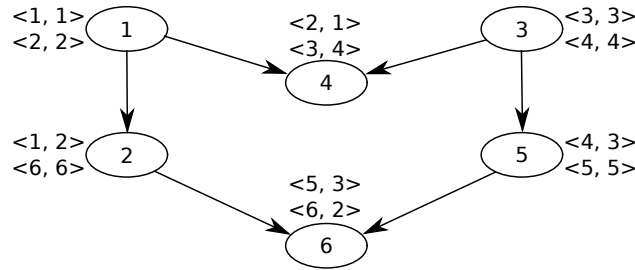

Figure 3: Final version of the rotation poset constructed from the sample in Table 5.

Note that in Section 4.2, we use this instance as a follow-up example. We give a complete example in the sense that we first introduce a sample SM

15

instance here, then we show how to construct this instance by starting from a special SAT formulation, without knowing anything about the SM instance.

Next, we define some subproblems of π_{ab}. We construct the proof for showing the problem π_{1b} is $\mathcal{NP}$-complete by first showing that its restricted version π_{11} is $\mathcal{NP}$-complete. In order to do this, we work on the specific family of Stable Marriage instances F, proving $\mathcal{NP}$-completeness for π_{11}^F. Then we generalize the results. The formal definitions of all three problems are given below.

Definition 2 (π_{1b}). *A particular case of π_{ab}. Decision problem for $(1, b)$-supermatch.*
INPUT: *$b \in \mathbb{N}$, and a Stable Marriage instance $\mathcal{I}$.*
QUESTION: *Is there a $(1, b)$-supermatch for $\mathcal{I}$?*

Definition 3 (π_{11}). *A particular case of π_{1b}. Decision problem for $(1, 1)$-supermatch.*
INPUT: *A Stable Marriage instance $\mathcal{I}$.*
QUESTION: *Is there a $(1, 1)$-supermatch for $\mathcal{I}$?*

Definition 4 (π_{11}^F). *A particular case of π_{11}. Decision problem for $(1, 1)$-supermatch for problem family F.*
INPUT: *A Stable Marriage instance $\mathcal{I} \in$ F.*
QUESTION: *Is there a $(1, 1)$-supermatch for $\mathcal{I}$?*

4.2. The Definition of SAT-SM

In this section, we define a special, restricted case of SAT that takes additional inputs. As mentioned before, this SAT formulation is inspired by the specific family F defined earlier. Please note that this definition is long and it requires some rules and conditions to be applied on the input of the problem. The intuition behind imposing these rules and conditions is to reflect the specified properties of the family F. This special SAT formulation is actually equivalent to finding $(1, 1)$-supermatches on F, which we prove later in the paper. However, for the moment, we would like to give some insight to the reader about why we need all the conditions and the rules in the definition of this formulation to make him/her able to follow it more easily.

All the Boolean and integer variables in the input of the problem are inspired by the rotations. First, we give an abstraction of the use of these variables. The set of integers corresponds to the set of rotations in the

16

rotation poset of a Stable Marriage instance $\mathcal{I}$. Assume that a matching M, where S is M's corresponding closed subset is a $(1, 1)$-supermatch to $\mathcal{I}$. The Boolean variables then correspond to the status of the rotations with respect to S. More specifically, for each rotation identified by a unique integer e, each y_e represents if the corresponding rotation e is a sink node in S, each s_e represents if e is included in S, p_e represents if all parents of e are in S but $e \notin S$.

Second, the conditions on the lists reflect the properties of F. **Rule 1** captures the property of the problem defined in Lemma 1. If **Rule 1** is not satisfied, then previously broken pairs might be re-matched which is not allowed in the SM [12].

Finally, the conditions on the clauses model the problem $\boldsymbol{\pi_{11}^F}$. In what is presented below, the purpose of the clauses in Ⓐ ensure each man is included either in a sink node in S or all of its parents are in S. Clauses in Ⓑ ensure the result is a closed subset. Clauses in Ⓒ define how a rotation can be identified as a sink node. Additionally, clauses in Ⓓ define how a rotation can be identified as having its parents in S but not itself.

The reason for including all these clauses and rules should become more clear to the reader when we are constructing a Stable Marriage instance from an instance of SAT-SM in the proof of Theorem 4. Now we begin our definition for the special case of SAT, namely SAT-SM.

Input: SAT-SM takes as input a set of integers $\boldsymbol{\mathcal{X}} = [1, |\boldsymbol{\mathcal{X}}|]$, $n \in \mathbb{N}^+$ lists $l_1, l_2, \ldots, l_n$ where each list is an ordered list of integers of $\boldsymbol{\mathcal{X}}$, and three sets of disjoint Boolean variables $Y = \{y_e \mid e \in \boldsymbol{\mathcal{X}}\}$, $S = \{s_e \mid e \in \boldsymbol{\mathcal{X}}\}$, and $P = \{p_e \mid e \in \boldsymbol{\mathcal{X}}\}\}$.

Conditions on the Lists: The lists $l_1, \ldots, l_n$ are subject to the following constraints. First, each list $\forall a \in [1, n]$, $l_a = (\mathcal{X}_1^a, \ldots, \mathcal{X}_{k_{l_a}}^a)$, where $k_{l_a} = |l_a| \geq 2$. Second, each element of $\boldsymbol{\mathcal{X}}$ appears in exactly two different lists. For illustration, the set $\boldsymbol{\mathcal{X}}$ represents the indexes of rotations and a list l_a represents the index of each rotation containing the man m_a. The order in l_a specifies the path in the rotation poset from the first rotation to the last one for a man m_a. The requirement for having each index in two different lists is related to Property 1. In addition to those two conditions, we have the following rule over the lists:

[**Rule 1**] For any $\mathcal{X}_i^m$ and $\mathcal{X}_j^m$ from the same list l_m where $m \in [1, n]$ and

$j > i$, there does not exist any sequence S that starts at χ_i^m and ends at χ_j^m constructed by iterating the two consecutive rules below:

α) given $\chi_e^a \in S$, the next element in S is χ_{e+1}^a, where $e + 1 \leq k_{l_a}$.

β) given $\chi_e^a \in S$, the next element in S is χ_f^b, where $\chi_e^a = \chi_f^b$, $a \neq b \in [1, n]$, and $1 \leq f \leq k_{l_b}$.

Before continuing to the conditions on the clauses, we would like to present an example where **Rule 1** is not satisfied. Consider three lists defined over $\boldsymbol{\mathcal{X}} = \{1, 2, 3\}$ as: $l_1 = (1, 3), l_2 = (1, 2), l_3 = (2, 3)$. We use the properties α and β to construct a sequence that is not allowed by the rule. Let the pair $\langle i, l_k \rangle$ denote the element i in list l_k, and we use $\rightarrow_p$ to show the property p that is applied to move from one pair to the other one in the sequence. For elements 1 and 3 in from the same list l_1, consider the sequence $S = \langle 1, l_1 \rangle \rightarrow_\beta \langle 1, l_2 \rangle \rightarrow_\alpha \langle 2, l_2 \rangle \rightarrow_\beta \langle 2, l_3 \rangle \rightarrow_\alpha \langle 3, l_3 \rangle \rightarrow_\beta \langle 3, l_1 \rangle$. This sequence starts and ends at the same list l_1, where 1 appears before 3 in l_1. The existence of this sequence causes the given three lists to be excluded from the SAT-SM.

Conditions on the clauses. The CNF that defines SAT-SM is a conjunction of four groups of clauses: Ⓐ, Ⓑ, Ⓒ and Ⓓ. The groups are subject to the following conditions:

Ⓐ: For any list $l_a = (\chi_1^a, \ldots, \chi_{k_{l_a}}^a)$, we have a disjunction between the Y-elements and the P-elements as $\bigvee_{i=1}^{k_{l_a}} (y_{\chi_i^a} \vee p_{\chi_i^a})$.

$$\text{Ⓐ is defined by } \bigwedge_{a=1}^{n} \left(\bigvee_{i=1}^{k_{l_a}} (\boldsymbol{y_{\chi_i^a}} \vee \boldsymbol{p_{\chi_i^a}}) \right). \tag{1}$$

Ⓑ: For any list $l_a = (\chi_1^a, \ldots, \chi_{k_{l_a}}^a)$, we have a conjunction of disjunctions between two S-elements with consecutive indexes as $\bigwedge_{i=1}^{k_{l_a}-1} (s_{\chi_i^a} \vee \neg s_{\chi_{i+1}^a})$.

$$\text{Ⓑ is defined by } \bigwedge_{a=1}^{n} \bigwedge_{i=1}^{k_{l_a}-1} (\boldsymbol{s_{\chi_i^a}} \vee \neg \boldsymbol{s_{\chi_{i+1}^a}}). \tag{2}$$

Ⓒ: This group of clauses is split in two. For any list $l_a = (\chi_1^a, \ldots, \chi_{k_{l_a}}^a)$, the first sub-group C_1 contains all the clauses defined by the logic formula

18

$\bigwedge_{i=1}^{k_{l_a}-1} y_{\chi_i^a} \rightarrow (s_{\chi_i^a} \wedge \neg s_{\chi_{i+1}^a})$. In CNF notation it leads to $\bigwedge_{i=1}^{k_{l_a}-1} (\neg y_{\chi_i^a} \vee s_{\chi_i^a}) \wedge (\neg y_{\chi_i^a} \vee \neg s_{\chi_{i+1}^a})$. Note that, C_1 also covers the special case, when $i = k_{l_a}$.

$$C_1 \text{ is defined by } \bigwedge_{a=1}^{n} \left(\bigwedge_{i=1}^{k_{l_a}} (\neg y_{\chi_i^a} \vee s_{\chi_i^a}) \wedge \bigwedge_{i=1}^{k_{l_a}-1} (\neg y_{\chi_i^a} \vee \neg s_{\chi_{i+1}^a}) \right). \quad (3)$$

The second sub-group C_2 has three specific cases according to the position of elements in the ordered lists. As fixed above, each element of $\boldsymbol{\chi}$ appears in exactly two different lists. Thus, for any $e \in \boldsymbol{\chi}$, there exists two lists l_a and l_b such that $\chi_i^a = \chi_j^b = e$, where $i \in [1, k_{l_a}]$ and $j \in [1, k_{l_b}]$. For each couple of elements of $\boldsymbol{\chi}$ denoted by (χ_i^a, χ_j^b) that are equal to the same value e, we define a clause with these elements and the next elements in their lists respecting the ordering: $s_{\chi_i^a} \rightarrow (y_{\chi_i^a} \vee s_{\chi_{i+1}^a} \vee s_{\chi_{j+1}^b})$. With a CNF notation it leads to: $(\neg s_{\chi_i^a} \vee y_{\chi_i^a} \vee s_{\chi_{i+1}^a} \vee s_{\chi_{j+1}^b})$.

We add the two specific cases where χ_i^a or χ_j^b, or both are the last elements of their ordered lists. The complete formula for the set of clauses C_2 for each two element (χ_i^a, χ_j^b) s.t. $\chi_i^a = \chi_j^b$ is:

$$C_2 \left[\begin{array}{ll} \bigwedge_{i \neq k_{l_a},\ j \neq k_{l_b}} & \neg s_{\chi_i^a} \vee y_{\chi_i^a} \vee s_{\chi_{i+1}^a} \vee s_{\chi_{j+1}^b} \\[2ex] \bigwedge_{i \neq k_{l_a},\ j = k_{l_b}} & \neg s_{\chi_i^a} \vee y_{\chi_i^a} \vee s_{\chi_{i+1}^a} \\[2ex] \bigwedge_{i = k_{l_a},\ j = k_{l_b}} & \neg s_{\chi_{k_{l_a}}^a} \vee y_{\chi_{k_{l_a}}^a} . \end{array} \right. \quad (4)$$

Ⓓ: Similar to C_2, for each couple of elements of $\boldsymbol{\chi}$ denoted by (χ_i^a, χ_j^b) equal to the same value e, there exists a clause with these elements and the previous elements in their lists respecting the ordering: $p_{\chi_i^a} \leftrightarrow (\neg s_{\chi_i^a} \wedge s_{\chi_{i-1}^a} \wedge s_{\chi_{j-1}^b})$. In CNF notation, it leads to:

$(\neg p_{\chi_i^a} \vee \neg s_{\chi_i^a}) \wedge (\neg p_{\chi_i^a} \vee s_{\chi_{i-1}^a}) \wedge (\neg p_{\chi_i^a} \vee s_{\chi_{j-1}^b}) \wedge (s_{\chi_i^a} \vee \neg s_{\chi_{i-1}^a} \vee \neg s_{\chi_{j-1}^b} \vee p_{\chi_i^a})$.

By generalizing the formula for any couple, and by adding the two cases where $\chi_i^{l_a}$, or $\chi_j^{l_b}$, or both are the first elements of their respective lists, the complete formula Ⓓ for each two element (χ_i^a, χ_j^b) s.t. $\chi_i^a = \chi_j^b = e$ is

described by:

$$
\text{\textcircled{D}} \quad
\begin{bmatrix}
\displaystyle\bigwedge_{i \neq 1,\; j \neq 1} (\neg p_{\chi_i^a} \vee s_{\chi_{i-1}^a}) \wedge (\neg p_{\chi_j^b} \vee s_{\chi_{j-1}^b}) \wedge \\[2mm]
(s_{\chi_i^a} \vee \neg s_{\chi_{i-1}^a} \vee \neg s_{\chi_{j-1}^b} \vee p_{\chi_i^a}) \\[2mm]
\displaystyle\bigwedge_{i = 1,\; j \neq 1} (\neg p_{\chi_j^b} \vee s_{\chi_{j-1}^b}) \wedge (s_{\chi_i^a} \vee \neg s_{\chi_{j-1}^a} \vee p_{\chi_i^a}) \\[2mm]
\displaystyle\bigwedge_{i = 1,\; j = 1} (s_{\chi_1^a} \vee p_{\chi_1^a}) \\[2mm]
\wedge \quad (\neg p_e \vee \neg s_e).
\end{bmatrix}
\tag{5}
$$

To conclude the definition, the full CNF formula of SAT-SM is
$\text{\textcircled{A}} \wedge \text{\textcircled{B}} \wedge C_1 \wedge C_2 \wedge \text{\textcircled{D}}$.

We introduce a sample SAT-SM instance as an example to demonstrate the methods described in the rest of this section. Table 5 introduces a SAT-SM instance consisting of 6 lists and a set of 6 integers $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$. Note that, these lists are constructed using the set $\mathcal{X}$ by following the *conditions on the lists* described on Page 17 defined for SAT-SM and **Rule 1**.

Each list can be seen as a man involved in the corresponding Stable Marriage instance $\mathcal{I}$. For instance, l_1 corresponds to m_1, l_2 corresponds to m_2, etc. Then, each element can be seen as a rotation in the rotation poset of $\mathcal{I}$, i.e. 1 corresponds to ρ_1, 2 corresponds to ρ_2, and so on. Then, this table describes which rotations involve which men. For example, m_1 is included in rotations ρ_1, ρ_2 and none of the others contain m_1.

Consider the rotation poset of the SM instance given in Figure 3. Note that the lists capture the men in rotations in the poset. For instance, m_1 is involved in the rotations ρ_1, ρ_2 and no others, etc.

Then, using the lists given in Table 5, clauses of the model can easily be derived by following the *conditions on the clauses* described on Page 18. The complete list of clauses are detailed in Table 6. Note that, although it is not specified on the table for making it easy to read, the SAT-SM instance is a conjunction of all the clauses in each group (i.e. $\text{\textcircled{A}} \wedge \text{\textcircled{B}} \wedge C_1 \wedge C_2 \wedge \text{\textcircled{D}}$). Also note that, this example instance is one of the smallest that can be created respecting the rules of the SAT-SM.

Table 5: An instance of SAT-SM of 6 lists and 6 integers.

List	Elements
l_1	(1, 2)
l_2	(1, 4)
l_3	(3, 4)
l_4	(3, 5)
l_5	(5, 6)
l_6	(2, 6)

Table 6: Clauses of the SAT-SM instance given in Table 5.

list	Ⓐ	Ⓑ	C_1
l_1	$(y_1 \vee y_2 \vee p_1 \vee p_2)$	$(s_1 \vee \neg s_2)$	$(\neg y_1 \vee s_1) \wedge (\neg y_2 \vee s_2) \wedge (\neg y_1 \vee \neg s_2)$
l_2	$(y_1 \vee y_4 \vee p_1 \vee p_4)$	$(s_1 \vee \neg s_4)$	$(\neg y_1 \vee s_1) \wedge (\neg y_4 \vee s_4) \wedge (\neg y_1 \vee \neg s_4)$
l_3	$(y_3 \vee y_4 \vee p_3 \vee p_4)$	$(s_3 \vee \neg s_4)$	$(\neg y_3 \vee s_3) \wedge (\neg y_4 \vee s_4) \wedge (\neg y_3 \vee \neg s_4)$
l_4	$(y_3 \vee y_5 \vee p_3 \vee p_5)$	$(s_3 \vee \neg s_5)$	$(\neg y_3 \vee s_3) \wedge (\neg y_5 \vee s_5) \wedge (\neg y_3 \vee \neg s_5)$
l_5	$(y_5 \vee y_6 \vee p_5 \vee p_6)$	$(s_5 \vee \neg s_6)$	$(\neg y_5 \vee s_5) \wedge (\neg y_6 \vee s_6) \wedge (\neg y_5 \vee \neg s_6)$
l_6	$(y_2 \vee y_6 \vee p_2 \vee p_6)$	$(s_2 \vee \neg s_6)$	$(\neg y_2 \vee s_2) \wedge (\neg y_6 \vee s_6) \wedge (\neg y_2 \vee \neg s_6)$

χ	C_2	Ⓓ
1	$(\neg s_1 \vee y_1 \vee s_2 \vee s_4)$	$(\neg p_1 \vee \neg s_1) \wedge (s_1 \vee p_1)$
2	$(\neg s_2 \vee y_2 \vee s_6)$	$(\neg p_2 \vee \neg s_2) \wedge (\neg p_2 \vee s_1) \wedge (s_2 \vee \neg s_1 \vee p_2)$
3	$(\neg s_3 \vee y_3 \vee s_4 \vee s_5)$	$(\neg p_3 \vee \neg s_3) \wedge (s_3 \vee p_3)$
4	$(\neg s_4 \vee y_4)$	$(\neg p_4 \vee \neg s_4) \wedge (\neg p_4 \vee s_1) \wedge (\neg p_4 \vee s_3) \wedge$ $(s_4 \vee \neg s_1 \vee \neg s_3 \vee s_4)$
5	$(\neg s_5 \vee y_5 \vee s_6)$	$(\neg p_5 \vee \neg s_5) \wedge (\neg p_5 \vee s_3) \wedge (s_5 \vee \neg s_3 \vee p_5)$
6	$(\neg s_6 \vee y_6)$	$(\neg p_6 \vee \neg s_6) \wedge (\neg p_6 \vee s_5) \wedge (\neg p_6 \vee s_2) \wedge$ $(s_6 \vee \neg s_5 \vee \neg s_2 \vee p_6)$

4.3. The Complexity of SAT-SM

We show some properties of the SAT-SM in this section to be used in our theorem.

Lemma 2. *There always exist clauses of minimum length 4 that are defined over positive literals in* Ⓐ*.*

Proof. For any list of ordered elements $l_a \in \{l_1, l_2, \ldots, l_n\}$, the length of each list is defined as $k_{l_a} \geq 2$ in SAT-SM, which results in Ⓐ having n clauses that have at least 4 positive literals in each. $\square$

Lemma 3. *There always exist clauses of length 2 that are defined over two negative literals in* Ⓒ.

Proof. The clauses in Ⓒ consists of C_1 and C_2. The clauses in C_1 are defined over the list of ordered elements. For any two consecutive elements in a list $l_a \in \{l_1, l_2, \ldots, l_n\}$, there exists one clause in C_1: $\bigwedge_{i=1}^{k_{l_a}-1}(\neg y_{\chi_i^a} \vee \neg s_{\chi_{i+1}^a})$. By definition, the minimum length of an ordered list l_a is $k_{l_a} = 2$ and therefore the formula contains at least n clauses defined over two negative literals. □

Lemma 4. *Any clause defined over only positive literals of size at least two is not affine.*

Proof. Any clause C of the given form with k positive literals has $2^k - 1$ valid assignments. The cardinality of an affine relation is always a power of 2 [23]. Thus, C is not affine. □

The SAT-SM problem is the question of finding an assignment of the Boolean variables that satisfies the above CNF formula.

Theorem 3. *The* SAT-SM *problem is* $\mathcal{NP}$-*complete.*

Proof. We use Schaefer's dichotomy theorem (Theorem 1) to prove that SAT-SM is $\mathcal{NP}$-complete [23]. Schaefer identifies six cases, where if any one of them is valid the SAT problem is solved in polynomial time. Any SAT formula that does not satisfy any of those 6 is $\mathcal{NP}$-complete.

It is easy to see the properties b, c, and f in Schaefer's Dichotomy do not apply to SAT-SM due to Lemma 2. Similarly, properties a and d are not satisfiable because of Lemma 3. The clauses in Ⓐ are defined as clauses over positive literals and it is known that they always exist by Lemma 2. By applying Lemma 4 on the clauses in Ⓐ, we infer that property e is not applicable either. Hence, SAT-SM is $\mathcal{NP}$-complete. □

We can now present the main result of the paper.

Theorem 4. *The decision problem* $\boldsymbol{\pi}_{\mathbf{11}}^{\boldsymbol{F}}$ *is* $\mathcal{NP}$-*complete.*

Proof. The witness is known to be polynomial-time decidable [7]. Therefore, $\boldsymbol{\pi}_{\mathbf{11}}^{\boldsymbol{F}}$ is in $\mathcal{NP}$. We show that $\boldsymbol{\pi}_{\mathbf{11}}^{\boldsymbol{F}}$ is $\mathcal{NP}$-complete by presenting a polynomial reduction from the SAT-SM problem to $\boldsymbol{\pi}_{\mathbf{11}}^{\boldsymbol{F}}$ as follows.
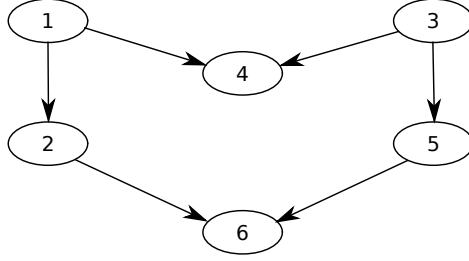
Figure 4: Initial version of the rotation poset constructed from the sample in Table 5.

From an instance $\mathcal{I}_{SSM}$ of SAT-SM, we construct in polynomial time an instance $\mathcal{I}$ of $\boldsymbol{\pi_{11}^F}$. This means the construction of the rotation poset $\Pi_F = (\mathcal{V}_F, E_F)$ with all stable pairs in the rotations, and the preference lists.

We begin by constructing the set of rotations $\mathcal{V}_F$ and then proceed by deciding which man is a part of which stable pair in which rotation. First, $\forall e \in \boldsymbol{\chi}$, we have a corresponding rotation ρ_e. Initially, each rotation contains two "empty" pairs. Second, $\forall l_a, a \in [1, n], \forall \chi_i^a \in [1, k_{l_a}]$, we insert m_a as the man to the first empty pair in rotation $\rho_{\chi_i^a}$. Since $k_{l_a} \geq 2$ from Lemma 2, each man of $\boldsymbol{\pi_{11}^F}$ is involved in at least two rotations (satisfying Property 4 of family F).

As each χ_i^a appears in exactly two different lists l_a and l_b, each rotation is guaranteed to contain exactly two pairs involving different men m_a, m_b (Property 1 of family F), and to possess at most two immediate predecessors and two immediate successors in Π_F (Property 2 of family F).

For the construction of the set of edges E_F, for each couple of elements of $\boldsymbol{\chi}$ denoted by (χ_i^a, χ_{i+1}^a), $a \in [1, n], \forall i \in [1, k_{l_a} - 1]$, we add an edge from $\rho_{\chi_i^a}$ to $\rho_{\chi_{i+1}^a}$. The shape of the rotation poset of the example given in Table 5 after the nodes and edges are created is shown in Figure 4. Note that the labeling of the nodes (or rotations) come from the integers in $\boldsymbol{\chi}$.

It is important to observe that this construction yields each edge in E_F representing a Type 1 relationship (Property 3 and Property 4 of family F). Since each edge links two rotations, where exactly one of the men is involved in both rotations. Now, in order to complete the rotation poset Π_F, the women involved in rotations must also be added. The following procedure is used to complete the rotation poset:

1. For each element $\chi_1^a \in \boldsymbol{\chi}$, with $a \in [1, n]$, let $\rho_{\chi_1^a}$ be the rotation that involves man m_a. In this case, the partner of m_a in $\rho_{\chi_1^a}$ is completed by

inserting woman w_a, so that the resulting rotation contains the stable pair $\langle m_a, w_a \rangle \in \rho_{\chi_1^a}$.

2. We perform a breadth-first search on the rotation poset from the completed rotations. For each complete rotation $\rho = (\langle m_i, w_b \rangle, \langle m_k, w_d \rangle) \in \mathcal{V}_F$, if the immediate successor of ρ involves m_i (resp. m_k), let ρ_{s1} (resp. ρ_{s2}) be the immediate successor of ρ that modifies m_i (resp. m_k). If ρ_{s1} exists, then we insert the woman w_d in ρ_{s1} as the partner of man m_i. In the same manner, if ρ_{s2} exists, we insert the woman w_b in ρ_{s2} as the partner of man m_k. The procedure creates at most two stable pairs $\langle m_i, w_d \rangle$ and $\langle m_k, w_b \rangle$ (see the illustration in Figure 2 (D)).

Remark. We now show that none of the constructed paths in the rotation poset cause a pair that was eliminated before by a rotation to be re-matched. **Rule 1** imposed on the SAT-SM ensures that there does not exist more than one path between any two rotations. Therefore, by imposing **Rule 1**, we can conclude that Lemma 1 is satisfied and none of the pairs that is eliminated by a preceding rotation is re-matched.

Throughout the construction we showed that all the properties required to have a valid rotation poset from the family F are satisfied. Using this process we are adding an equal number of women and men into the rotation poset. Figure 3, presented in the previous section, is the result of adding the men and the women in the rotations to Figure 4 using the procedure defined above.

The last step to obtain an instance $\mathcal{I}$ of $\boldsymbol{\pi_{11}^F}$ is the construction of the preference lists. Recall that $\Pi_F = (\mathcal{V}_F, E_F)$ is the rotation poset that we are building. By using the rotation poset Π_F created above, we can construct incomplete preference lists for the men and women. Gusfield et al. defined a procedure to show that every finite poset corresponds to a stable marriage instance [11]. They describe a method to create the preference lists using the rotation poset. We use a similar approach for creating the lists as detailed below:

- Apply topological sort on $\mathcal{V}_F$.

- For each man $m_i \in [1, n]$, insert woman w_i as the most preferred to m_i's preference list.

- For each woman $w_i \in [1, n]$, insert man m_i as the least preferred to w_i's preference list.

Table 7: The incomplete preference lists derived from the rotation poset in Figure 3.

m_1	1 2 6		w_1	3 2 1
m_2	2 1 4		w_2	5 6 1 2
m_3	3 4 1		w_3	6 5 4 3
m_4	4 3 5		w_4	2 3 4
m_5	5 3 2		w_5	4 5
m_6	6 2 3		w_6	1 6

- For each rotation $\rho \in \mathcal{V}_F$ in the ordered set, for each pair $\langle m_i, w_j \rangle$ produced by ρ, insert w_j to the man m_i's list in decreasing order of preference ranking. Similarly, place m_i to w_j's list in increasing order of preference ranking.

Lemma 1 on our rotation poset clearly states that each preference list contains each member of the opposite sex at most once. To finish, one can observe that the instance obtained respects the Stable Marriage requirements and the specific properties from problem family F.

Applying the procedure above, the preference lists of the men and women are found as shown in Table 7. The integers in the preference list of a man m_i denote the index of women. For instance, man m_1 prefers w_1 over w_2 and his least preferred partner is w_6. It is also similar for the preference lists of the women. Observe that, the lists obtained after the procedure correspond to the preference lists provided in Table 4, where the non-stable pairs are removed.

Having seen the construction of an SM instance $\mathcal{I}$ of $\boldsymbol{\pi_{11}}$ from a given instance $\mathcal{I}_{SSM}$ of SAT-SM we now present the equivalence between the two decision problems: $\boldsymbol{\pi_{11}^F}$ and SAT-SM.
$\Rightarrow$ Suppose that there exists a solution to an instance $\mathcal{I}$ of the decision problem $\boldsymbol{\pi_{11}^F}$. Then we have a $(1,1)$-supermatch and its corresponding closed subset S. As defined in Section 2, $\mathrm{L}(S)$ is the set of sink nodes of the graph induced by the rotations in S, $\mathrm{N}(S)$ the set of nodes such that all their predecessors (if any) are in S but not themselves. From these two sets, we can assign all the literals in $\mathcal{I}_{SSM}$ as follows:

- For each rotation $\rho_i \in \mathrm{L}(S)$, set $y_i = true$. Otherwise, set $y_i = false$.

- For each rotation $\rho_i \in S$, set $s_i = true$. Otherwise, set $s_i = false$.

- For each rotation $\rho_i \in N(S)$, set $p_i = true$. Otherwise, set $p_i = false$.

If S represents a $(1, 1)$-supermatch, that means by removing only one rotation present in $L(S)$ or by only adding one rotation from $N(S)$, any pair of the corresponding stable matching can be repaired with no additional modifications. Thus every men must be contained in a sink or a neighbor node. This leads to having for each man one of the literals assigned to true in his list in SAT-SM. Therefore every clause in Ⓐ in Equation 1 are satisfied.

For the clauses in Ⓑ in Equation 2, for any man's list the clauses are forcing each s_i literal to be true if the next one s_{i+1} is. By definition of a closed subset, from any sink of S, all the preceding rotations (integers in the lists) must be in S. And thus every clause in Ⓑ is satisfied.

As the clauses in Ⓒ altogether capture the definition of being a sink node of the graph induced by the rotations in S, they are all satisfied by $L(S)$. At last, for the clauses in Ⓓ, it is also easy to see that any rotation being in $N(S)$ is equivalent to not being in the solution and having predecessors in. Thus all the clauses are satisfied. Thus we can conclude that this assignment satisfies the SAT formula of $\mathcal{I}_{SSM}$.

In terms of example, the closed subset $S = \{\rho_1, \rho_2, \rho_3\}$ corresponds to one of the $(1, 1)$-supermatches in its solution set. Assuming S is the solution by π_{11}^F, the transformation of the solution to the solution of SAT-SM is found as in Table 8.

Table 8: Solution transformation.

true	false
$s_1, s_2, s_3, y_2, y_3, p_4, p_5$	$s_4, s_5, s_6, y_1, y_4, y_5, y_6, p_1, p_2, p_3, p_6$

$\Leftarrow$ Suppose that there exists a solution to an instance $\mathcal{I}_{SSM}$ of the decision problem SAT-SM. Thus we have a valid assignment to satisfy the SAT formula of $\mathcal{I}_{SSM}$. We construct a closed subset S to solve $\mathcal{I}$. As previously, we use the sets $L(S)$ and $N(S)$, then for each literal y_i assigned to true, we put the rotation ρ_i in $L(S)$. We do the same for p_i and s_i as above.

The clauses in Ⓑ enforce the belonging to S of all rotations preceding any element of S, thus the elements in S form a closed subset. To obtain a $(1, 1)$-supermatch, we have to be sure we can repair any couple by removing only one rotation present in $L(S)$ or by only adding one rotation from $N(S)$. The clauses in Ⓒ enforce the rotations in $L(S)$ to be without successors in

S. And in the same way the clauses in Ⓓ enforce the rotations in $N(S)$ to not be in S but have their predecessors in the solution.

Now we just have to check that all the men are contained in at least one rotation from $\mathrm{L}(S) \cup \mathrm{N}(S)$. From the clauses from Ⓐ, we know that at least one y_e or p_e for any man m_i is assigned to true. Thus from this closed subset S, we can repair any couple $\langle m_i, w_j \rangle$ using one modification by removing/adding the rotation having m_i. Since there exists a $1-1$ equivalence between a stable matching and the closed subset in the rotation poset, we have a $(1,1)$-supermatch. □

Corollary 1. *Both decision problems $\boldsymbol{\pi_{11}}$ and $\boldsymbol{\pi_{1b}}$ are $\mathcal{NP}$-complete.*

Note that, we cannot generalize our results to $\boldsymbol{\pi_{ab}}$. Because there is no known polynomial-time witness for (a,b)-supermatches for $a > 1$.

5. Threshold and Polynomial Cases

We present a family of instances for which a $(1,1)$-supermatch can be found in polynomial time. Then, we show that $(2,0)$-supermatches do not exist. Last, we discuss the existence of $(a,0)$-supermatches.

5.1. Polynomial Cases

We first introduce a labelling for rotations in the rotation poset. We assume that the label of each source node of the rotation poset is 1, and for every other rotation, the rotations label is calculated by the number of edges to its furthest predecessor summed by the predecessor's label. Note that the furthest predecessor is always a source node due to having a poset. Formally, the label of $\rho \in \mathcal{V}$, denoted by $l(\rho)$, is defined recursively as: $l(\rho) = 1$ if ρ is a source node, and $l(\rho) = 1 + \max_{\rho' \in N^-(\rho)}(l(\rho'))$ otherwise. The set of nodes that have the same label l defines a *level*, referred as *Level l*. The purpose of the labelling process is to be able to find some levels in the rotation poset such that every level corresponds to a unique stable matching.

Lemma 5. *All rotations that have the same label are incomparable.*

Proof. The proof is derived from the definition of the labelling function. If one rotation precedes another, i.e. they are comparable, the successors label must be larger than the predecessors label. Therefore, if two rotations have the same label then they are incomparable. □

Using the intuition of the levels, we define a family of SM instances called F_w. Each SM instance in this family is identified by having a Level l such that the set of all rotations of size 2 in Level l combined with the rotations in the Level $l+1$ involve all the non-fixed men of the underlying instance. Then, we show that all SM instances in this family has a $(1, 1)$-supermatch and we can find this $(1, 1)$-supermatch by a polynomial-time algorithm. We often encounter with these instances when trying to create SM instances that have as many stable matchings as possible. Thus, these instances are interesting in the sense that some large SM instances always contain a $(1, 1)$-supermatch and this stable matching is easy to find.

In Definition 5, we give a formal definition of F_w.

Definition 5. F_w *is a family of Stable Marriage instances such that each instance has a stable matching M, where the M and its corresponding closed subset S have the two following properties:*
1) All the sink nodes of size two in S have the same label;
2) $|X(R^*(L(S) \cup N(S)))| = n$, where n denotes the number of non-fixed men; i.e. the rotations of size two, contained in the union of the sink nodes and neighbor nodes of S, cover all the non-fixed men.

Lemma 6. *All instances of F_w admit a $(1, 1)$-supermatch.*

Proof. Proposition 2 and Property 2 from Definition 5 lead directly to this result. ☐

Lemma 7. *There exists a $(1, 1)$-supermatch M in any instance $\mathcal{I}$ of F_w such that in M's closed subset S all the sink nodes of size two have the same label l and all the nodes in level l are in S.*

Proof. Let M be such a $(1, 1)$-supermatch for $\mathcal{I}$. Let L be the set of sink nodes of S. Assume there exists a set of nodes L^*, where $L \subseteq L^*$ such that all the nodes in L^* have the label l. Then, the closed subset S^* defined with $L \cup L^*$ as sink nodes corresponds to a $(1, 1)$-supermatch. The proof is quite straightforward. We know from Proposition 2 that $|X(R^*(L(S) \cup N(S)))| = n$ (where n is the number of non-fixed men). Moreover, we have $R^*(L(S^*) \cup N(S^*))$ is a superset of $R^*(L(S) \cup N(S))$. Hence, $|X(R^*(L(S^*) \cup N(S^*)))| = n$ and by Proposition 2, the stable matching corresponding to S^* is a $(1, 1)$-supermatch.

In short, if a subset of rotations in a level corresponds to a $(1, 1)$-supermatch with the given properties, the closed subset that covers all the rotations in

that level also corresponds to a $(1,1)$-supermatch. Thus, in order to find a $(1,1)$-supermatch for $\mathcal{I}$, it is sufficient to look for the closed subset whose set of sink nodes contains all the rotations in a level. $\qquad\square$

Definition 6 (π_{11}^w). *A particular case of π_{11} for problem family* F_w.
INPUT*: A Stable Marriage instance $\mathcal{I} \in F_w$.*
OUTPUT*: Find a $(1,1)$-supermatch for $\mathcal{I}$?*

We now state our tractability result.

Theorem 5. π_{11}^w *is solvable in* $O(|\mathcal{V}| + |E|)$ *time.*

Proof. Consider an instance $\mathcal{I}$ in F_w. The principle of our polynomial-time procedure for any instance of F_w is to first identify the different levels, then to look for the closed subset whose set of sink nodes contains all the rotations in every level to see if it corresponds to a $(1,1)$-supermatch (until one is found). The levels in the rotation poset can be identified by applying topological sort on the rotation poset first and then by calculating the labels of rotations in the sorted list. Topological sorting results in a list of nodes such that if rotation ρ' is preceding rotation ρ in Π, then ρ' appears before ρ in the sorted list. The running time of topological sort is $O(|\mathcal{V}| + |E|)$.

Let N_l denote the set of all rotations at Level l, and N_{l+1} denote the ones at level $l+1$. Then, for any Level l, if the union set of all rotations of size 2 in Level l and Level $l+1$ contains all the non-fixed men, i.e. $|X(R^*(N_l \cup N_{l+1}))| = n$, then l corresponds to a $(1,1)$-supermatch. The corresponding stable matching is constructed by adding all predecessors of all the rotations at Level l, including the rotations themselves into the closed subset. Note that, the last Level l corresponds to a $(1,1)$-supermatch only if $|X(R^*(N_l))| = n$.

Let $m < |V|$ be the number of levels in the poset. The check at a level to see if it contains all the men can be done in constant time, and construction of the corresponding stable matching is $O(|\mathcal{V}|)$, which is computed only once at the end. Therefore, the running time of the overall algorithm is $O(|\mathcal{V}| + |E|)$ due to the Topological sorting.

Also, observe that the number of stable matchings this algorithm creates is bounded by the number of levels, where the number of levels is equal to the maximum number of edges in a path between a source node and a sink node. $\qquad\square$

Table 9: Preference lists for men (left) and women (right) for an instance $\mathcal{I}_F$ of size 8.

m_1	8 6 2 5 1 4 7 3	w_1	4 8 7 6 1 5 2 3
m_2	5 7 8 2 1 4 3 6	w_2	4 2 7 1 5 3 8 6
m_3	2 1 7 5 6 8 3 4	w_3	2 1 5 3 7 6 4 8
m_4	4 3 2 6 7 5 8 1	w_4	8 4 3 6 5 7 2 1
m_5	4 1 6 5 2 7 8 3	w_5	4 2 6 7 8 1 5 3
m_6	5 1 8 4 3 2 7 6	w_6	7 4 1 3 8 5 6 2
m_7	7 2 3 8 4 5 6 1	w_7	5 4 1 3 6 7 8 2
m_8	3 5 7 4 2 1 6 8	w_8	3 8 5 1 4 2 6 7

Note that, although the algorithm described in the proof of Theorem 5 is complete for the family F_w, it does not guarantee a solution to instances that are not in F_w. We give an example of such a Stable Marriage instance. That is, using the notion of levels of a rotation poset fails to find a $(1, 1)$-supermatch.

In Table 9, a Stable Marriage instance $\mathcal{I}_F$ of size 8, that does not belong to the family F_w is given. Figure 5 is the rotation poset that represents this instance, respectively.
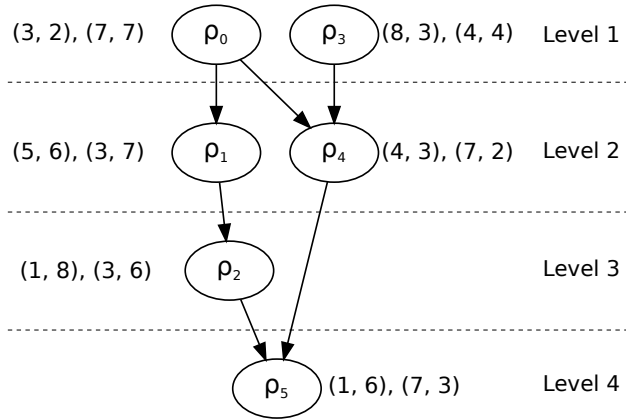

Figure 5: Rotation poset of the instance $\mathcal{I}_F$ given in Table 9.

Recall that if a man has the same partner in all stable matchings of the underlying instance then the man is said to be fixed. However, if he has at least one alternative partner, then the man is non-fixed. The instance $\mathcal{I}_F$ contains two fixed men (m_2 and m_6) and 12 stable matchings. Only one stable match-

30

ing, namely $M = \{(1,8),(2,5),(3,6),(4,3),(5,7),\ (6,1),(7,2),(8,4)\}$, is a $(1,1)$-supermatch. The closed subset corresponding to M is $S = \{\rho_0,\rho_1,\rho_3\}$. Let N_i denote a level identified by the algorithm described earlier as follows: $N_1 = \{\rho_0,\rho_3\}$, $N_2 = \{\rho_1,\rho_4\}$, $N_3 = \{\rho_2\}$, and $N_4 = \{\rho_5\}$. Additionally, the corresponding closed subsets for each level is as follows: $S_{N_1} = \{\rho_0,\rho_3\}$, $S_{N_2} = \{\rho_0,\rho_3,\rho_1,\rho_4\}$, $S_{N_3} = \{\rho_0,\rho_1,\rho_2\}$, $S_{N_4} = \{\rho_0,\rho_3,\rho_1,\rho_4,\rho_2,\rho_5\}$. We can calculate the $(1,b)$-robustness of each stable matching corresponding to these levels. The stable matching of the sets S_{N_1}, S_{N_2}, and S_{N_3} are all $(1,2)$-supermatches, whereas the stable matching of S_{N_4} is a $(1,3)$-supermatch.

5.2. Finding an $(a,0)$-Supermatch

We consider the case of finding an $(a,0)$-supermatch and show that for the base case, $(2,0)$-supermatches need not exist in general. Let $\boldsymbol{\pi_{a0}}$ denote the problem of finding an $(a,0)$-supermatch.

Definition 7 ($\boldsymbol{\pi_{a0}}$). *A particular case of $\boldsymbol{\pi_{ab}}$. Decision problem for $(a,0)$-supermatch.*
INPUT: *A Stable Marriage instance $\mathcal{I}$.*
QUESTION: *Is there an $(a,0)$-supermatch for $\mathcal{I}$?*

Firstly note that $(1,0)$-supermatches do not exist because in order to find new partners for a couple, at least one other couple must break up. We next show that $(2,0)$-supermatches need not exist in general and then discuss the generic method for solving $\boldsymbol{\pi_{a0}}$.

Theorem 6. *Given any Stable Marriage instance where the number of non-fixed men n is at least 3, there do not exist any $(2,0)$-supermatches.*

Proof. Suppose that a $(2,0)$-supermatch M exists and let S be its closed subset. We argue that the only way to repair M (if two couples decide to break up) is to obtain a closed subset by either adding a rotation of size 2 to S or to remove a rotation of size 2 from S. Let m_1, m_2, m_3 be three distinct non-fixed men (recall that $n > 2$). For every pair of men $\{m_1,m_2\},\{m_2,m_3\},\{m_1,m_3\}$, there exists a rotation ρ of size 2 that involves both men such that $\rho \in L(S)$ or $\rho \in N(S)$. Therefore, there necessarily exists a man $m \in \{m_1,m_2,m_3\}$ that is involved in two rotations ρ_1 and ρ_2 that are both in $L(S)$ or both in $N(S)$. Since ρ_1 and ρ_2 involve m, then they are comparable. This contradicts the fact that $L(S)$ (respectively $N(S)$) contains rotations that are incomparable. $\qquad\square$

We make the following observation regarding the general case of $(a, 0)$-supermatches. Let n denote the number of non-fixed men in a stable marriage instance $\mathcal{I}$ such that $2 < a \geq n$. Also, for a given stable matching M and a number of unwanted pairs in M, let the term *repair matching* denote the closest stable matching to M, where the unwanted pairs do not exist in the repair matching. Suppose that M is a $(a, 0)$-supermatch for $\mathcal{I}$. This means that it is possible to find a repair stable matching to M for a breakage involving every combination of non-fixed men of size a. Considering Theorem 6, we have the intuition that $(a, 0)$-supermatches need not exist in general. However, if they exist, we suspect they exist in instances that have many number of stable matchings.

5.3. An Overview of the Complexity Results

Figure 6 summarizes the hierarchy between different cases of finding an (a, b)-supermatch.
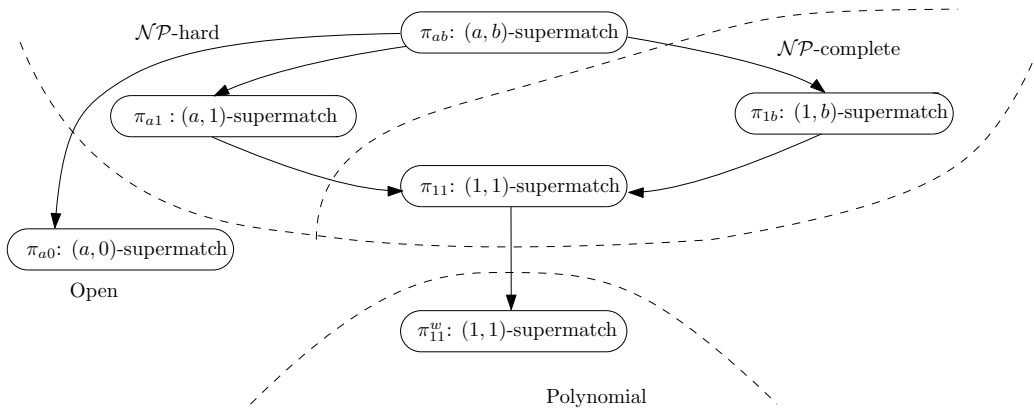

Figure 6: Illustration of the complexity hierarchy between the different cases of RSM.

6. Concluding Remarks

We study the complexity of the Robust Stable Marriage (RSM) problem by introducing a special case of SAT that is $\mathcal{NP}$-complete, and show a transformation from this formulation to Stable Marriage problem. We find that deciding if there exists a $(1, 1)$-supermatch is $\mathcal{NP}$-complete. The case of $(1, b)$-supermatch is also $\mathcal{NP}$-complete. However, for the more general case of (a, b)-supermatch, we do not know yet whether it is a member of $\mathcal{NP}$ or

not. Therefore, we can only say that it is $\mathcal{NP}$-hard by generalization. We also identify a number of polynomial-time solvable families of Stable Marriage instances. We think it is interesting to find other families, as they give great insight to the structure of the problem. Although we show that $(2,0)$-supermatches do not exist, we do not have a generalization to the $(a,0)$-case.

Some interesting directions for future work in this area include: resolving the complexity of the general (a,b)-supermatches case (by finding a verification procedure or proving that none exists); proving if $(a,0)$-supermatches exist or do not exist; searching for (a,b)-supermatches in the context of other matching problems, etc. Note that all of our methods are based on rotations and the rotation poset. Therefore, studying the rotation posets of the stable matching problems in more detail or finding procedures that are not based on rotations could be interesting.

Acknowledgements

References

[1] Afacan, M. O., 2012. Group robust stability in matching markets. Games and Economic Behavior 74 (1), 394–398.

[2] Aziz, H., Biró, P., Gaspers, S., de Haan, R., Mattei, N., Rastegari, B., 2016. Stable matching with uncertain linear preferences. CoRR abs/1607.02917.
URL http://arxiv.org/abs/1607.02917

[3] Barber, F., Salido, M. A., Sep. 2015. Robustness, stability, recoverability, and reliability in constraint satisfaction problems. Knowl. Inf. Syst. 44 (3), 719–734.
URL http://dx.doi.org/10.1007/s10115-014-0778-3

[4] Dantsin, E., Hirsch, E. A., 2009. Worst-case upper bounds. In: Handbook of Satisfiability. pp. 403–424.
URL https://doi.org/10.3233/978-1-58603-929-5-403

[5] Drummond, J., Boutilier, C., 2013. Elicitation and approximately stable matching with partial preferences. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI '13. AAAI Press, pp. 97–105.
URL http://dl.acm.org/citation.cfm?id=2540128.2540145

[6] Gale, D., Shapley, L. S., 1962. College admissions and the stability of marriage. The American Mathematical Monthly 69 (1), 9–15.

[7] Genc, B., Siala, M., Simonin, G., O'Sullivan, B., 2017. Finding robust solutions to stable marriage. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. IJCAI '17.

[8] Genc, B., Siala, M., Simonin, G., O'Sullivan, B., 2017. On the complexity of robust stable marriage. In: Combinatorial Optimization and Applications - 11th International Conference, COCOA 2017, Shanghai, China, December 16-18, 2017, Proceedings, Part II. pp. 441–448.

[9] Genc, B., Siala, M., Simonin, G., O'Sullivan, B., 2017. Robust stable marriage. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 4925–4926.
URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14785

[10] Ginsberg, M. L., Parkes, A. J., Roy, A., 1998. Supermodels and robustness. In: In AAAI/IAAI. pp. 334–339.

[11] Gusfield, D., Irving, R., Leather, P., Saks, M., 1987. Every finite distributive lattice is a set of stable matchings for a small stable marriage instance. Journal of Combinatorial Theory, Series A 44 (2), 304–309.

[12] Gusfield, D., Irving, R. W., 1989. The Stable Marriage Problem: Structure and Algorithms. MIT Press, Cambridge, MA, USA.

[13] Hebrard, E., 2007. Robust solutions for constraint satisfaction and optimisation under uncertainty. Ph.D. thesis, University of New South Wales.

[14] Hebrard, E., Hnich, B., Walsh, T., 2004. Robust solutions for constraint satisfaction and optimization. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious

Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004. pp. 186–190.

[15] Hebrard, E., Hnich, B., Walsh, T., 2004. Super solutions in constraint programming. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, First International Conference, CPAIOR 2004, Nice, France, April 20-22, 2004, Proceedings. pp. 157–172.

[16] Irving, R. W., Leather, P., Aug. 1986. The complexity of counting stable marriages. SIAM J. Comput. 15 (3), 655–667.
URL http://dx.doi.org/10.1137/0215048

[17] Jacobovic, R., 2016. Perturbation robust stable matching. CoRR abs/1612.08118.

[18] Kojima, F., 2011. Robust stability in matching markets. Theoretical Economics 6 (2), 257–267.

[19] Mai, T., Vazirani, V. V., 2018. Finding stable matchings that are robust to errors in the input. CoRR abs/1804.00553.
URL http://arxiv.org/abs/1804.00553

[20] Manlove, D., 2013. Algorithmics Of Matching Under Preferences. Theoretical computer science. World Scientific Publishing.
URL http://eprints.gla.ac.uk/79820/

[21] Menon, V., Larson, K., 2018. Robust and approximately stable marriages under partial information. CoRR abs/1804.09156.
URL http://arxiv.org/abs/1804.09156

[22] Salido, M. A., Barber, F., Ingolotti, L. P., 2012. Robustness for a single railway line: Analytical and simulation methods. Expert Syst. Appl. 39 (18), 13305–13327.

[23] Schaefer, T. J., 1978. The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. STOC '78. ACM, New York, NY, USA, pp. 216–226.
URL http://doi.acm.org/10.1145/800133.804350

[24] Soyster, A. L., 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21 (5), 1154–1157.
    URL http://www.jstor.org/stable/168933

[25] Woodward, J., 2006. Some varieties of robustness. Journal of Economic Methodology 13 (2), 219–240.
    URL https://doi.org/10.1080/13501780600733376

[26] Xu, H., Caramanis, C., Mannor, S., Dec. 2009. Robustness and regularization of support vector machines. J. Mach. Learn. Res. 10, 1485–1510.
    URL http://dl.acm.org/citation.cfm?id=1577069.1755834

[27] Xu, Y., Huang, G., Qin, X., Cao, M., 2009. Srccp: A stochastic robust chance-constrained programming model for municipal solid waste management under uncertainty. Resources, Conservation and Recycling 53 (6), 352 – 363.
    URL http://www.sciencedirect.com/science/article/pii/S0921344909000329