



On the Complexity of Robust Stable Marriage

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

► To cite this version:

Begum Genc, Mohamed Siala, Gilles Simonin, Barry O ' Sullivan. On the Complexity of Robust Stable Marriage. COCOA 2017 11th Annual International Conference on Combinatorial Optimization and Applications , Dec 2017, Shanghai, China. hal-01689884

HAL Id: hal-01689884

<https://hal.science/hal-01689884>

Submitted on 22 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Complexity of Robust Stable Marriage

Begum Genc¹, Mohamed Siala¹, Gilles Simonin², Barry O’Sullivan¹

¹ Insight, Centre for Data Analytics, Department of Computer Science, University
College Cork, Ireland

`{begum.genc, mohamed.siala, barry.osullivan}@insight-centre.org`,

² TASC, Institut Mines Telecom Atlantique, LS2N UMR 6004, Nantes, France
`gilles.simonin@imt-atlantique.fr`

Abstract. *Robust Stable Marriage (RSM)* is a variant of the classical *Stable Marriage* problem, where the robustness of a given stable matching is measured by the number of modifications required for repairing it in case an unforeseen event occurs. 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 also the partners of at most b others. In order to show deciding if there exists an (a, b) -supermatch is \mathcal{NP} -complete, we first introduce a SAT formulation that is \mathcal{NP} -complete by using Schaefer’s Dichotomy Theorem. Then, we show the equivalence between the SAT formulation and finding a $(1, 1)$ -supermatch on a specific family of instances.

1 Introduction

Matching under preferences is a multidisciplinary family of problems, mostly studied by the researchers in the field of economics and computer science. There are many variants of the matching problems such as College Admission, Hospital/Residents, Stable Marriage, Stable Roommates, etc. The reader is referred to the book written by Manlove for a comprehensive background on the subject [1].

Current studies in the literature indicate that different robustness notions for different matching problems are being studied. To the best of our knowledge, the very first robustness notion in matching problems is studied on the Geometric Stable Roommates problem [2]. Later on, there appear a few research papers on robustly stable mechanisms in matching markets [3–5]. The most recent notions are proposed in Stable Marriage problem, where one of them uses a probability model and a social cost function to measure robustness [6], and the other one uses a cost function to calculate repair costs of each stable matching and use it as a measure of robustness [7].

We work on the robustness notion of stable matching proposed by Genc et. al. [7]. In the context of Stable Marriage, the purpose is to find a matching M between men and women such that no pair $\langle man, woman \rangle$ prefer each other to their situations in M . The authors of [7] introduced the notion of (a, b) -supermatch

as a measure of robustness. An (a, b) -*supermatch* is a stable matching such that if any a agents (men or woman) break up it is possible to find another stable matching by changing the partners of those a agents with also changing the partners of at most b others. This notion is inspired by the work of Ginsberg et al. on (a, b) -supermodels in Boolean Satisfiability [8], and (a, b) -super solutions in CSP by Hebrard et.al. [9–11]. Both finding an (a, b) -supermodel and an (a, b) -super solution are shown to be \mathcal{NP} -complete. However, they leave the complexity of this problem open [7].

The focus of this paper is to study the complexity of finding an (a, b) -supermatch. In order to show that the general case of RSM, which is the decision of existence of an (a, b) -supermatch, is \mathcal{NP} -complete, it is sufficient to show that a restricted version of the general problem is \mathcal{NP} -complete. Thus, we first show that the decision problem for finding a $(1, 1)$ -supermatch on a restricted family of instances is \mathcal{NP} -complete, then we generalize this complexity result to the general case.

Figure 1 illustrates the hierarchy between different cases of finding an (a, b) -supermatch.

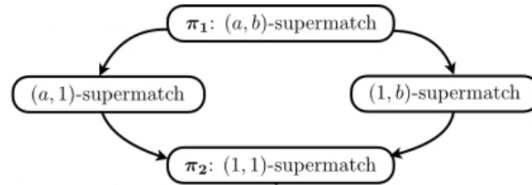


Fig. 1. Illustration of the complexity hierarchy between the different cases of RSM.

In Section 2, the notations and the basics of the Stable Marriage problem are introduced, then we refer to some theorems that are already present in the literature. In Section 3, we describe a restricted family of Stable Marriage instances. In Section 4, after defining a specific SAT formulation based on the properties of the restricted family, we show by reduction that the decision version of finding an $(1, 1)$ -supermatch is \mathcal{NP} -complete.

2 Notations & Background

An instance of the *Stable Marriage problem (with incomplete lists)* takes as input a set of men $U = \{m_1, m_2, \dots, m_{n_1}\}$ and a set of women $W = \{w_1, w_2, \dots, w_{n_2}\}$ where each person has an ordinal preference list over members of the opposite sex. For the sake of simplicity we suppose in the rest of the 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) 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 prefers w_j to $M(m_i)$ and w_j 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 rest of the 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 $\Psi \subset M$ of a stable pairs that are not fixed, there exists a stable matching M' such that $M' \cap \Psi = \emptyset$ and $d(M, M') - a \leq b$ [7].

Definition 1 (π_1) *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. A **rotation** $\rho = (\langle m_{k_0}, w_{k_0} \rangle, \langle m_{k_1}, w_{k_1} \rangle, \dots, \langle m_{k_{l-1}}, w_{k_{l-1}} \rangle)$ (where $l \in \mathbb{N}^*$) is an ordered list of pairs in 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 $\langle m_{k_i}, w_{k_i} \rangle$ is *eliminated* by ρ , whereas $\langle m_{k_i}, w_{k_{i+1}} \rangle$ is *produced* by ρ , and that ρ is *exposed* on 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 m_i is *involved* in ρ . There exists a partial order for rotations. A rotation ρ' is said to precede another rotation ρ (denoted by $\rho' \prec \rho$), 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]. Note that this relation is transitive, that is, $\rho'' \prec \rho' \wedge \rho' \prec \rho \implies \rho'' \prec \rho$. Two rotations are said to be *incomparable* if none of them precede the other.

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 vertex 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 $\langle m_i, w_j \rangle$ is in 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 . 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 *leaf node* and a node that has no incoming edges is called *root 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 . Every closed subset in the rotation poset corresponds to a stable matching [12]. Let $L(S)$ be the set of rotations that are the leaf 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 sub-graphs, 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 leaf nodes of Π_1 and $N(S)$ corresponds to the root nodes of Π_2 .

An important remark is that there is a 1-1 correspondence between the matchings in \mathcal{T} and the sets of incomparable rotations in V . 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 $L(S)$.

Predecessors of a rotation ρ in a rotation poset are denoted by $N^-(\rho)$ and successors are denoted by $N^+(\rho)$. We also denote by $X(R)$ the set of men involved in a set of rotations R .

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 . Figure 2 represents the rotation poset and all the rotations associated with this sample.

m_0	0 6 5 2 4 1 3	w_0	2 1 6 4 5 3 0
m_1	6 1 4 5 0 2 3	w_1	0 4 3 5 2 6 1
m_2	6 0 3 1 5 4 2	w_2	2 5 0 4 3 1 6
m_3	3 2 0 1 4 6 5	w_3	6 1 2 3 4 0 5
m_4	1 2 0 3 4 5 6	w_4	4 6 0 5 3 1 2
m_5	6 1 0 3 5 4 2	w_5	3 1 2 6 5 4 0
m_6	2 5 0 6 4 3 1	w_6	4 6 2 1 3 0 5

Table 1. Preference lists for men (left) and women (right) for a sample instance of size 7.

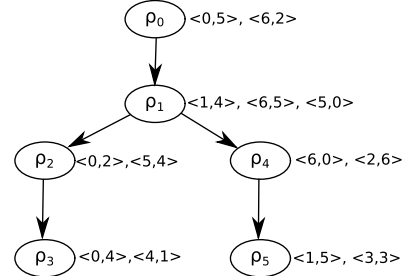


Table 2. Rotation poset of the instance given in Table 1.

All the stable matchings of the instance given in Figure 2 are listed in Table 3. For the sake of 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 on M_1 , also denoted by $M_2 = M_1/\rho_1 = \{(0, 2), (1, 5), (2, 6), (3, 3), (4, 1), (5, 4), (6, 0)\}$. For M_2 , the leaf and neighbor nodes can be identified as $L(S_2) = \{\rho_1\}$ and $N(S_2) = \{\rho_2, \rho_4\}$.

2.1 Schaefer's Dichotomy Theorem for Satisfiability

The original Schaefer's dichotomy theorem is proposed in [13]. In this section, we use the same terminology and notations as in [14]. A **literal** is a Boolean

Stable Matching	Pairs
M_0	$\{(0, 5), (1, 4), (2, 6), (3, 3), (4, 1), (5, 0), (6, 2)\}$
$M_1 = M_0/\rho_0$	$\{(0, 2), (1, 4), (2, 6), (3, 3), (4, 1), (5, 0), (6, 5)\}$
$M_2 = M_1/\rho_1$	$\{(0, 2), (1, 5), (2, 6), (3, 3), (4, 1), (5, 4), (6, 0)\}$
$M_3 = M_2/\rho_4$	$\{(0, 2), (1, 5), (2, 0), (3, 3), (4, 1), (5, 4), (6, 6)\}$
$M_4 = M_3/\rho_5$	$\{(0, 2), (1, 3), (2, 0), (3, 5), (4, 1), (5, 4), (6, 6)\}$
$M_5 = M_2/\rho_2$	$\{(0, 4), (1, 5), (2, 6), (3, 3), (4, 1), (5, 2), (6, 0)\}$
$M_6 = M_5/\rho_4 = M_3/\rho_2$	$\{(0, 4), (1, 5), (2, 0), (3, 3), (4, 1), (5, 2), (6, 6)\}$
$M_7 = M_4/\rho_2 = M_6/\rho_5$	$\{(0, 4), (1, 3), (2, 0), (3, 5), (4, 1), (5, 2), (6, 6)\}$
$M_8 = M_5/\rho_3$	$\{(0, 1), (1, 5), (2, 6), (3, 3), (4, 4), (5, 2), (6, 0)\}$
$M_9 = M_6/\rho_3 = M_8/\rho_4$	$\{(0, 1), (1, 5), (2, 0), (3, 3), (4, 4), (5, 2), (6, 6)\}$
$M_{10} = M_7/\rho_3 = M_9/\rho_5$	$\{(0, 1), (1, 3), (2, 0), (3, 5), (4, 4), (5, 2), (6, 6)\}$

Table 3. The list of all stable matchings for the instance given in Figure 2.

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 a positive (respectively negative) literal. A *linear equation over the 2-element field* is an expression of the form $x_1 \oplus x_2 \dots \oplus x_k = \delta$ 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 $\phi : \{true, false\}^k \rightarrow \{true, false\}$. Let (x_1, \dots, x_k) be a sequence of Boolean variables and ϕ be a Boolean constraint of arity k . The pair $\langle \phi, (x_1, \dots, x_k) \rangle$ is called a *constraint application*. An assignment A to (x_1, \dots, x_k) satisfies $\langle \phi, (x_1, \dots, x_k) \rangle$ 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 A .

Let \mathcal{C} be a set of Boolean constraints. **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 [14, 13]. 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 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 predecessors and at most 2 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. Note that, the ordering of the pairs is not important as there exist only two pairs in each rotation.

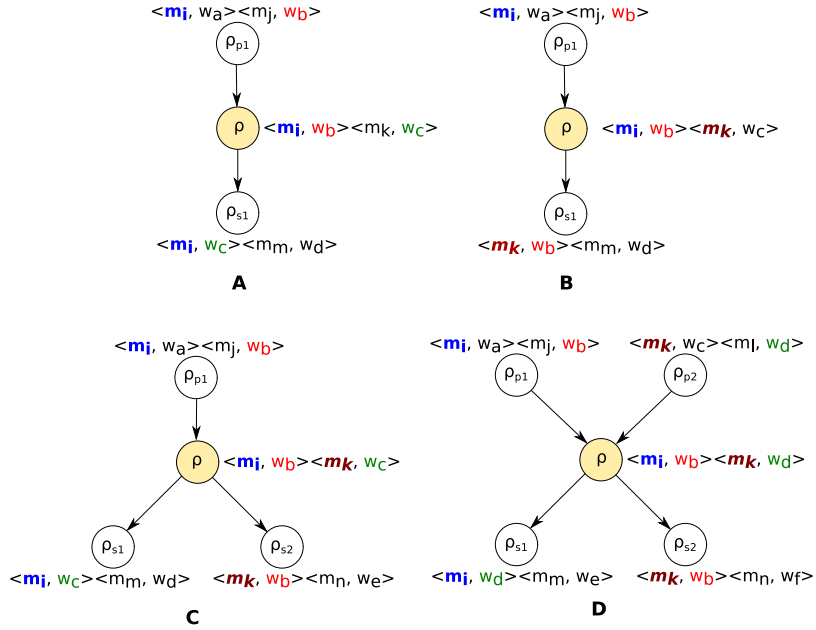


Fig. 2. An illustration of the men/women included in rotations for cases where a rotation has exactly 1 predecessor and 1 successor (A,B), 1 predecessor and 2 successors, which is also similar to 1 successor and 2 predecessors (C), 2 predecessors and 2 successors (D).

We would like to remark the difference between the cases A and B in Figure 2. Due to Property 3, any two rotations that have an edge between, contain a man and a woman in common. When the case is generalized to 3 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 the 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. *For each 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 contradiction that m_e is involved in all rotations on P_1 and w_f is involved in all rotations on P_2 . This scenario is likely to occur as the Property 3 indicates that all edges are of type 1, which is also easy to observe on the Figure 2, 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, the 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 can not be produced again.

Definition 2 (π_1^F) *A particular case of π_1 , with the restrictions from problem family F.*

Definition 3 (π_2) *The special case of π_1 , where $a = 1, b = 1$.*

Definition 4 (π_2^F) *INPUT: A Stable Marriage instance \mathcal{I} from family F.
QUESTION: Is there a (1,1)-supermatch for \mathcal{I} ?*

4 Complexity results

In order to prove that the general problem π_1 is \mathcal{NP} -complete, we show that the restricted family problem π_1^F is \mathcal{NP} -complete. And in order to do this, we first prove it for a particular case noted π_2^F by introducing an \mathcal{NP} -complete SAT formulation denoted by SAT-SM.

4.1 \mathcal{NP} -completeness

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

Conditions on the lists: The lists l_1, \dots, l_n are subject to the following constraints: First, each list $\forall a \in [1, n]$, $l_a = (\chi_1^a, \dots, \chi_{k_{l_a}}^a)$, where $k_{l_a} = |l_a| \geq 2$. Second, each element of \mathcal{X} appears in exactly two different lists. For illustration, the set \mathcal{X} represents the indexes of rotations and a list l_a represents the index of each rotation having 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 . And the restriction 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 χ_i^m and χ_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 σ and θ below:

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

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

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

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

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

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

(C): This group of clauses is split in two. For any list $l_a = (\chi_1^a, \dots, \chi_{k_{l_a}}^a)$, the first sub-group C_1 contains all the clauses defined by the logic formula $\bigwedge_{i=1}^{k_{l_a}-1} y_{\chi_i^a} \rightarrow s_{\chi_i^a} \wedge \neg s_{\chi_{i+1}^a}$. With a 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 \mathcal{X} appears in exactly two different lists. Thus, for any $e \in \mathcal{X}$, 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 \mathcal{X} 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:

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

Ⓓ: Similarly to C_2 , for each couple of elements of \mathcal{X} 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: $p_{\chi_i^a} \leftrightarrow \neg s_{\chi_i^a} \wedge s_{\chi_{i-1}^a} \wedge s_{\chi_{j-1}^b}$. With a CNF notation, it leads to:

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

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:

$$\textcircled{\text{D}} \left[\begin{array}{ll} \bigwedge_{i \neq 1, j \neq 1} & (\neg p_{\chi_i^a} \vee s_{\chi_{i-1}^a}) \wedge (\neg p_{\chi_j^b} \vee s_{\chi_{j-1}^b}) \wedge \\ & (s_{\chi_i^a} \vee \neg s_{\chi_{i-1}^a} \vee \neg s_{\chi_{j-1}^b} \vee p_{\chi_i^a}) \\ \bigwedge_{i = 1, j \neq 1} & (\neg p_{\chi_j^b} \vee s_{\chi_{j-1}^b}) \wedge (s_{\chi_i^a} \vee \neg s_{\chi_{j-1}^b} \vee p_{\chi_i^a}) \\ \bigwedge_{i = 1, j = 1} & s_{\chi_1^a} \vee p_{\chi_1^a} \\ \wedge & \neg p_e \vee \neg s_e \end{array} \right. \quad (5)$$

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

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

Proof. For any list of ordered elements $l_a \in \{l_1, l_2, \dots, l_n\}$, 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 some clauses of length 2 that are defined over two negative literals in \textcircled{C} .*

Proof. The clauses in \textcircled{C} consists of two groups. The first group is defined over the list of ordered elements. For any two consecutive elements in a list $l_a \in \{l_1, l_2, \dots, l_n\}$, there exists two clauses: $\bigvee_{i=1}^{k_{l_a}-1} (\neg y_{\chi_i^a} \vee s_{\chi_i^a}) \wedge (\neg y_{\chi_i^a} \vee \neg s_{\chi_{i+1}^a})$. By definition, the minimum length of an ordered list l_a is $k_{l_a} = 2$ and therefore the minimum-length list yields in 2 clauses of the defined form. Therefore, the first group includes $2 \times \sum_{a \in \{1, \dots, n\}} (k_{l_a} - 1)$ clauses. \square

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 have $2^k - 1$ valid assignments. The cardinality of an affine relation is always a power of 2 [13]. Thus, C is not affine. \square

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

Theorem 2. *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 [13]. 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 a , d , and f in Schaefer's Dichotomy do not apply to SAT-SM due to Lemma 2. Similarly, properties b and c are not satisfiable because of Lemma 3. The clauses in \textcircled{A} 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 \textcircled{A} , we infer that property e is not applicable either. Hence, SAT-SM is \mathcal{NP} -complete. \square

We can now present the main result of the paper.

Theorem 3. *The decision problem π_2^F is \mathcal{NP} -complete.*

Proof. The verification is shown to be polynomial-time decidable [7]. Therefore, π_2^F is in \mathcal{NP} . We show that π_2^F is \mathcal{NP} -complete by presenting a polynomial reduction from the SAT-SM problem to π_2^F as follows.

From an instance \mathcal{I}_{SSM} of SAT-SM, we construct in polynomial time an instance \mathcal{I} of π_2^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 first start 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, $\forall e \in \mathcal{X}$, 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 π_2^F is involved in at least two rotations (satisfying Property 4).

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), and to possess at most two predecessors and two successors in Π_F (Property 2).

For the construction of the set of arcs E_F , for each couple of elements of \mathcal{X} denoted by (χ_i^a, χ_{i+1}^a) , $a \in [1, n], \forall i \in [1, k_{l_a} - 1]$, we add an arc from $\rho_{\chi_i^a}$ to $\rho_{\chi_{i+1}^a}$. Note that this construction, yields in each arc in E representing a type 1 relationship (Property 3 and Property 4). Because each arc 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 $\chi_1^a \in \mathcal{X}$, 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 $\rho = (\langle m_i, w_b \rangle, \langle m_k, w_d \rangle) \in \mathcal{V}_F$, let ρ_{s1} (resp. ρ_{s2}) be one of the successor of ρ and modifying 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). From the fact that each woman w_b appears in the next rotation as partnered with the next man of the current rotation ρ , in the SAT-SM definition it is equivalent to going from χ_y^i to χ_{z+1}^k on lists where $\chi_y^i = \chi_z^k, y \in [1, n], z \in [1, n - 1]$. Thus the path where the woman appears follow a sequence defined as the one in [Rule 1] from the SAT-SM definition. By this rule, we can conclude that Lemma 1 is satisfied.

All along 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 equal number of women and men in the rotation poset.

The last step to obtain an instance \mathcal{I} of π_2^F is the construction of the preference lists. By using the rotation poset created above, we can construct incomplete preference lists for the men and women. Gusfield et. al. define a procedure to show that every finite poset corresponds to a stable marriage instance [15]. In their work, 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.

- 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.

The Lemma 1 imposed on our rotation poset clearly involves 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.

\Leftarrow Suppose that there exists a solution to an instance \mathcal{I} of the decision problem π_2^F . Then we have a $(1,1)$ -supermatch and its corresponding closed subset S . As defined in Section 2, $L(S)$ is the set of leaf nodes of the graph induced by the rotations in S , $N(S)$ the set of nodes such that all their predecessors 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 $\rho_i \in L(S)$, set $y_i = \text{true}$. Otherwise, set $y_i = \text{false}$.
- For each rotation $\rho_i \in S$, set $s_i = \text{true}$. Otherwise, set $s_i = \text{false}$.
- For each rotation $\rho_i \in N(S)$, set $p_i = \text{true}$. Otherwise, set $p_i = \text{false}$.

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 any men must be contained in a leaf 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 \textcircled{A} in Equation 1 are satisfied. Therefore every clause in \textcircled{A} are satisfied.

For the clauses in \textcircled{B} 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. For the clauses in \textcircled{B} , 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 leaf of S , all the preceding rotations (indexes in the lists) must be in S . And thus every clause in \textcircled{B} is satisfied.

As the clauses in \textcircled{C} altogether capture the definition of being a leaf node of the graph induced by the rotations in S , they are all satisfied by $L(S)$. At last, for the clauses in \textcircled{D} , 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 satisfy the SAT formula of \mathcal{I}_{SSM} .

\Rightarrow 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 are doing the same for p_i and s_i as above.

The clauses in \textcircled{B} 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 \textcircled{C} enforce the rotations in $L(S)$ to be without successors in S . And in the same way the clauses in \textcircled{D} 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 $L(S) \cup N(S)$. By the clauses from \textcircled{A} , 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 $\langle m_i, w_j \rangle$ in 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. \square

Corollary 1 *From the Theorem 3 and by generality, both decision problems π_1 and π_2 are \mathcal{NP} -complete.*

5 Concluding Remarks

We study the complexity of the Robust Stable Marriage (RSM) problem. In order to show that given a Stable Marriage instance, deciding if there exists an (a, b) -supermatch is \mathcal{NP} -complete, we first introduce a SAT formulation which models a specific family of Stable Marriage instances. We show that the formulation is \mathcal{NP} -complete by Schaefer’s Dichotomy Theorem. Then we apply a reduction from this problem to prove the \mathcal{NP} -completeness of RSM.

6 Acknowledgements

This research has been funded by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

1. Manlove, D.: Algorithmics Of Matching Under Preferences. Theoretical computer science. World Scientific Publishing (2013)
2. Arkin, E.M., Bae, S.W., Efrat, A., Okamoto, K., Mitchell, J.S., Polishchuk, V.: Geometric stable roommates. Information Processing Letters **109**(4) (2009) 219 – 224
3. Kojima, F.: Robust stability in matching markets. Theoretical Economics **6**(2) (2011) 257–267
4. Afacan, M.O.: Group robust stability in matching markets. Games and Economic Behavior **74**(1) (2012) 394–398
5. Drummond, J., Boutilier, C.: Elicitation and approximately stable matching with partial preferences. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. IJCAI ’13, AAAI Press (2013) 97–105
6. Jacobovic, R.: Perturbation robust stable matching. CoRR **abs/1612.08118** (2016)

7. Genc, B., Siala, M., Simonin, G., O'Sullivan, B.: Finding robust solutions to stable marriage. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. IJCAI '17 (2017)
8. Ginsberg, M.L., Parkes, A.J., Roy, A.: Supermodels and robustness. In: In AAAI/IAAI. (1998) 334–339
9. Hebrard, E., Hnich, B., Walsh, T.: 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. (2004) 157–172
10. Hebrard, E., Hnich, B., Walsh, T.: Robust solutions for constraint satisfaction and optimization. In: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22–27, 2004. (2004) 186–190
11. Hebrard, E.: Robust solutions for constraint satisfaction and optimisation under uncertainty. PhD thesis, University of New South Wales (2007)
12. Gusfield, D., Irving, R.W.: The Stable Marriage Problem: Structure and Algorithms. MIT Press, Cambridge, MA, USA (1989)
13. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. STOC '78, New York, NY, USA, ACM (1978) 216–226
14. Dantsin, E., Hirsch, E.A.: Worst-case upper bounds. In: Handbook of Satisfiability. (2009) 403–424
15. Gusfield, D., Irving, R., Leather, P., Saks, M.: Every finite distributive lattice is a set of stable matchings for a small stable marriage instance. Journal of Combinatorial Theory, Series A **44**(2) (1987) 304 – 309