

1 **A Structural and Algorithmic Study of**
2 **Stable Matching Lattices of "Nearby" Instances,**
3 **with Applications**

4 **Rohith Reddy Gangam** 
5 University of California, Irvine, Irvine, CA, USA

6 **Tung Mai** 
7 Adobe Research, San Jose, CA, USA

8 **Nitya Raju** 
9 University of California, Irvine, Irvine, CA, USA

10 **Vijay V. Vazirani** 
11 University of California, Irvine, Irvine, CA, USA

12 **Abstract**

13 Recently [18] identified and initiated work on a new problem, namely understanding structural
14 relationships between the lattices of solutions of two “nearby” instances of stable matching. They
15 also gave an application of their work to finding a *robust stable matching*. However, the types of
16 changes they allowed in going from instance A to B were very restricted, namely any one agent
17 executes an *upward shift*.

18 In this paper, we allow any one agent to permute its preference list *arbitrarily*. Let M_A and M_B
19 be the sets of stable matchings of the resulting pair of instances A and B , and let \mathcal{L}_A and \mathcal{L}_B be
20 the corresponding lattices of stable matchings. We prove that the matchings in $M_A \cap M_B$ form a
21 sublattice of both \mathcal{L}_A and \mathcal{L}_B and those in $M_A \setminus M_B$ form a join semi-sublattice. These properties
22 enable us to obtain a polynomial time algorithm for not only finding a stable matching in $M_A \cap M_B$,
23 but also for obtaining the partial order, as promised by Birkhoff’s Representation Theorem [7]. As a
24 result, we can generate all matchings in this sublattice.

25 Our algorithm also helps solve a version of the robust stable matching problem. We discuss another
26 potential application, namely obtaining new insights into the incentive compatibility properties of
27 the Gale-Shapley Deferred Acceptance Algorithm.

28 **2012 ACM Subject Classification** Theory of computation → Algorithmic game theory and mechanism
29 design

30 **Keywords and phrases** stable matching, robust solutions, finite distributive lattice, Birkhoff’s
31 Representation Theorem

32 **Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2022.30

33 **Related Version** *Full Version:* <https://arxiv.org/pdf/1804.05537.pdf>

34 **Funding** *Vijay V. Vazirani:* This work was supported in part by NSF grant CCF-1815901.

35 **1 Introduction**

36 The seminal 1962 paper of Gale and Shapley [14] introduced the stable matching problem
37 and gave the Deferred Acceptance (DA) Algorithm for it. In the process, they initiated the
38 field of matching-based market design. Over the years, numerous researchers unearthed the
39 remarkably deep and pristine structural properties of this problem – this led to polynomial
40 time algorithms for numerous problems, in particular those addressing various operations
41 related to the lattice of stable matchings, see details below as well as in the books [17, 15,
42 20, 22, 12].



© Rohith Reddy Gangam, Tung Mai, Nitya Raju and Vijay V. Vazirani;

licensed under Creative Commons License CC-BY 4.0

42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science
(FSTTCS 2022).

Editors: Anuj Dawar and Venkatesan Guruswami; Article No. 30; pp. 30:1–30:20

 Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

30:2 A Study of Stable Matching Lattices of "Nearby" Instance

43 Recently [18] identified and initiated work on a new problem which appears to be fundamental and deserving of an in-depth study, namely understanding structural relationships
44 between the lattices of solutions of two "nearby" instances. [18] had given an application
45 of their work to finding a *robust stable matching* as described below. Let us say that two
46 instance A and B of stable matching are *nearby instances* if B is obtained from A when
47 one agent changes their preference list. Such pairs of instances arise naturally in an even
48 more important context: the study of incentive compatibility of the DA Algorithm: one
49 of the agents manipulates its preference list in order to get a better match. The types of
50 manipulations allowed in [18] were very restricted, namely any one agent executes an *upward
51 shift*, see definition below. They left the open problem of tackling more general changes.
52

53 [21] showed that finding a stable matching across k (≥ 2) arbitrary instances is NP-Hard.
54 In this paper, we allow any one agent to permute its preference list *arbitrarily*. Let A and B
55 be the resulting pair of instances, let M_A and M_B be the sets of their stable matchings and
56 \mathcal{L}_A and \mathcal{L}_B be the corresponding lattices of stable matchings. We prove that the matchings
57 in $M_A \cap M_B$ form a sublattice of both \mathcal{L}_A and \mathcal{L}_B and those in $M_A \setminus M_B$ form a join
58 semi-sublattice, see definitions in Section 1.1. This enables us to obtain a polynomial time
59 algorithm for not only finding a stable matching in $M_A \cap M_B$, but also to obtain the partial
60 order, promised by Birkhoff's Representation Theorem [7], which helps generate all matchings
61 in this sublattice. We also apply our algorithm to a more general setting for robust stable
62 matching than the one given in [18].

63 The setting defined in [18] was the following: Let A be an instance of stable matching on
64 n workers and n firms. A *domain of errors*, D , is defined via an operation called *upward shift*:
65 For a firm f , assume its preference list in instance A is $\{\dots, w_1, w_2, \dots, w_k, w, \dots\}$. Move
66 up the position of worker w so f 's list becomes $\{\dots, w, w_1, w_2, \dots, w_k, \dots\}$. An analogous
67 operation is defined on a worker w 's list; again some firm f on its list is moved up. For each
68 firm and each worker, consider all possible shifts to get the domain D ; clearly, $|D| = \binom{2n}{1} \binom{n}{2} =$
69 $O(n^3)$. Assume that *one error* is chosen from D via a given discrete probability distribution
70 over D to obtain instance B . A *robust stable matching* is a matching that is stable for A and
71 maximizes the probability of being stable for B . A polynomial time algorithm was given for
72 finding such a matching.

73 Since we allow an *arbitrary permutation* to be applied to any *one worker or any one
74 firm's* preference list, our domain of errors, say T , has size $2n(n!)$. Let $S \subseteq T$ and define a
75 *fully robust stable matching* w.r.t. S to be a matching that is stable for A and for *each* of the
76 $|S|$ instances obtained by introducing one error from S . We give an $O(|S|p(n))$ algorithm to
77 determine if such a matching exists and if so to find one, where p is a polynomial function.
78 In particular, if S is polynomial sized, then our algorithm runs in polynomial time. Clearly,
79 this notion is weaker than the previous one, since we cannot extend it to the probabilistic
80 setting; we leave that as an open problem, see Section 8.

81 In case all errors in S are on one side only, say the firms, it turns out that Algorithm D,
82 which is a simple modification of the Deferred Acceptance Algorithm, works; this algorithm
83 is given in Appendix D. However, extending this algorithm to the case that errors occur on
84 both sides, workers and firms, results in an algorithm (Algorithm D) that has exponential
85 runtime. Our polynomial time algorithm follows from a study of the sublattices of the lattice
86 of stable matchings.

87 Conway, see [17], proved that the set of stable matchings of an instance forms a finite
88 distributive lattice; see definitions in Section 2.2. Knuth [17] asked if every finite distributive
89 lattice is isomorphic to the lattice arising from an instance of stable matching. A positive
90 answer was provided by Blair [8]; for a much better proof, see [15]. A key fact about such

91 lattices is Birkhoff's Representation Theorem [7], which has also been called *the fundamental*
 92 *theorem for finite distributive lattices*, e.g., see [23]. It states that corresponding to such a
 93 lattice, \mathcal{L} , there is a partial order, say Π , such that \mathcal{L} is isomorphic to $L(\Pi)$, the lattice of
 94 closed sets of Π (see Section 2.2 for details). We will say that Π *generates* \mathcal{L} .

95 The following important question arose in the design of our algorithm: For a specified
 96 sublattice \mathcal{L}' of \mathcal{L} , obtain partial order Π' from Π such that Π' generates \mathcal{L}' . Our answer to
 97 this question requires a study of Birkhoff's Theorem from this angle; we are not aware of any
 98 previous application of Birkhoff's Theorem in this manner. We define a set of operations
 99 called compressions; when a compression is applied to a partial order Π , it yields a partial
 100 order Π' on (weakly) fewer elements. The following implication of Birkhoff's Theorem is
 101 useful for our purposes:

102 **Theorem 1.** *There is a one-to-one correspondence between the compressions of Π and the*
 103 *sublattices of $L(\Pi)$ such that if sublattice \mathcal{L}' of $L(\Pi)$ corresponds to compression Π' , then \mathcal{L}'*
 104 *is generated by Π' .*

105 A proof of Theorem 1, using stable matching lattices, is given in Section B for completeness.
 106 In the case of stable matchings, Π can be defined using the notion of *rotations*; see Section
 107 2.2 for a formal definition. Since the total number of rotations of a stable matching instance
 108 is at most $O(n^2)$, Π has a succinct description even though \mathcal{L} may be exponentially large.
 109 Our main algorithmic result is:

110 **Theorem 2.** *There is an algorithm for checking if there is a fully robust stable matching*
 111 *w.r.t. any set $S \subseteq T$ in time $O(|S|p(n))$, where p is a polynomial function. Moreover, if the*
 112 *answer is yes, the set of all such matchings forms a sublattice of \mathcal{L} and our algorithm finds a*
 113 *partial order that generates it.*

114 The importance of the stable matching problem lies not only in its efficient computability
 115 but also its good incentive compatibility properties. In particular, Dubins and Freedman
 116 [11] proved that the DA Algorithm is *dominant-strategy incentive compatible (DSIC)* for the
 117 proposing side. This opened up the use of this algorithm in a host of highly consequential
 118 applications, e.g., matching students to public schools in big cities, such as NYC and Boston,
 119 see [3, 1, 2]. In this application, the proposing side is taken to be the students; clearly, their
 120 best strategy is to report preference lists truthfully and not waste time and effort on “gaming”
 121 the system. In Section 8 we give a hypothetical situation regarding incentive compatibility
 122 in which Theorem 2 plays a role.

1.1 Overview of structural and algorithmic ideas

124 We start by giving a short overview of the structural facts proven in [18]. Let A and B be
 125 two instances of stable matching over n workers and n firms, with sets of stable matchings
 126 \mathcal{M}_A and \mathcal{M}_B , and lattices \mathcal{L}_A and \mathcal{L}_B , respectively. Let Π be the poset on rotations such
 127 that $L(\Pi) = \mathcal{L}_A$; in particular, for a closed set S , let $M(S)$ denote the stable matching
 128 corresponding to S . It is easy to see that if B is obtained from A by changing (upshifts
 129 only) the lists of only one side, either workers or firms, but not both, then the matchings in
 130 $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice of each of the two lattices (Proposition 6). Furthermore, if B
 131 is obtained by applying a shift operation, then $\mathcal{M}_{A \setminus B} = \mathcal{M}_A \setminus \mathcal{M}_B$ is also a sublattice of \mathcal{L}_A .
 132 Additionally, there is at most one rotation, ρ_{in} , that leads from $\mathcal{M}_A \cap \mathcal{M}_B$ to $\mathcal{M}_{A \setminus B}$ and at
 133 most one rotation, ρ_{out} , that leads from $\mathcal{M}_{A \setminus B}$ to $\mathcal{M}_A \cap \mathcal{M}_B$; moreover, these rotations can
 134 be found in polynomial time. Finally, for a closed set S of Π , $M(S)$ is stable for instance B

30:4 A Study of Stable Matching Lattices of "Nearby" Instance

135 iff $\rho_{\text{in}} \in S \Rightarrow \rho_{\text{out}} \in S$.

136 With a view to extending the results of [18], we consider the following abstract question.
137 Suppose instance B is such that $\mathcal{M}_A \cap \mathcal{M}_B$ and $\mathcal{M}_{A \setminus B}$ are both sublattices of \mathcal{L}_A , i.e., \mathcal{M}_A
138 is partitioned into two sublattices. Then, is there a polynomial time algorithm for finding a
139 matching in $\mathcal{M}_A \cap \mathcal{M}_B$? Our answer to this question is built on the following structural fact:
140 There exists a sequence of rotations $r_0, r_1, \dots, r_{2k}, r_{2k+1}$ such that a closed set of Π generates
141 a matching in $\mathcal{M}_A \cap \mathcal{M}_B$ iff it contains r_{2i} but not r_{2i+1} for some $0 \leq i \leq k$ (Proposition
142 19). Furthermore, this sequence of rotations can be found in polynomial time (see Section
143 4). Our generalization of Birkhoff's Theorem described in the Introduction is an important
144 ingredient in this algorithm. At this point, we do not know of any concrete error pattern,
145 beyond shift, for which this abstract setting applies.

146 Next, we address the case that $\mathcal{M}_{A \setminus B}$ is not a sublattice of \mathcal{L}_A . We start by proving that
147 if B is obtained by permuting the preference list of any one worker, then $\mathcal{M}_{A \setminus B}$ must be a
148 join semi-sublattice of \mathcal{L}_A (Lemma 31); an analogous statement holds if the preference list of
149 any one firm is permuted. Hence we study a second abstract question, namely lattice \mathcal{L}_A is
150 partitioned into a sublattice and a join semi-sublattice (see Section 5). These two abstract
151 questions are called **Setting I** and **Setting II**, respectively, in this paper.

152 For Setting II, we characterize a compression that yields a partial order Π' , such that
153 Π' generates the sublattice consisting of matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ (Theorem 20). We also
154 characterize closed sets of Π such that the corresponding matchings lie in this sublattice;
155 however, the characterization is too elaborate to summarize succinctly (see Proposition 25).
156 Edges forming the required compression can be found in polynomial time (Theorem 29),
157 hence leading to an efficient algorithm for finding a matching in $\mathcal{M}_A \cap \mathcal{M}_B$.

158 Finally, consider the setting given in the Introduction, with T being the super-exponential
159 set of all possible errors that can be introduced in instance A and $S \subset T$. We show that
160 the set of all matchings that are stable for A and for each of the instances obtained by
161 introducing one error from S forms a sublattice of \mathcal{L} and we obtain a compression of Π that
162 generates this sublattice (Section 7.2). Each matching in this sublattice is a fully robust
163 stable matching. Furthermore, given a weight function on all worker-firm pairs, we can
164 obtain, using the algorithm of [19], a maximum (or minimum) weight fully robust stable
165 matching.

166 2 Preliminaries

167 2.1 The stable matching problem and the lattice of stable matchings

168 The stable matching problem takes as input a set of workers $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ and a
169 set of firms $\mathcal{F} = \{f_1, f_2, \dots, f_n\}$; each agent has a complete preference ranking over the set
170 of opposite side. A matching M is a one-to-one correspondence between \mathcal{W} and \mathcal{F} . For
171 each pair $wf \in M$, w is called the partner of f in M (or M -partner) and vice versa. For a
172 matching M , a pair $wf \notin M$ is said to be *blocking* if they prefer each other to their partners.
173 A matching M is *stable* if there is no blocking pair for M .

174 Let M and M' be two stable matchings. We say that M *dominates* M' , denoted by
175 $M \preceq M'$, if every worker weakly prefers his partner in M to M' . Define the relation
176 *predecessor* as the transitive closure of dominates. The set of stable matchings forms a finite
177 distributive lattice under the above definition of predecessor. The lattice contains a matching,
178 M_0 , that dominates all others and a matching M_z that is dominated by all others. M_0 is
179 called the *worker-optimal matching*, since in it, each worker is matched to his most favorite
180 firm among all stable matchings. Similarly, M_z is *firm-optimal matching*.

181 2.2 Birkhoff's Theorem and rotations

182 It is easy to see that the family of closed sets (also called lower sets, Definition 5) of a partial
 183 order, say Π , is closed under union and intersection and forms a distributive lattice, with
 184 join and meet being these two operations, respectively; let us denote it by $L(\Pi)$. Birkhoff's
 185 theorem [7], states that corresponding to any finite distributed lattice, \mathcal{L} , there is a partial
 186 order, say Π , whose lattice of closed sets $L(\Pi)$ is isomorphic to \mathcal{L} , i.e., $\mathcal{L} \cong L(\Pi)$. We will
 187 say that Π generates \mathcal{L} .

188 One way to define the partial orders generating stable matching lattices is using the
 189 concept of *rotation*. For a worker w let $s_M(w)$ denote the first firm f on w 's list such that f
 190 strictly prefers w to her M -partner. Let $next_M(w)$ denote the partner in M of firm $s_M(w)$.
 191 A *rotation ρ exposed* in M is an ordered list of pairs $\{w_0f_0, w_1f_1, \dots, w_{r-1}f_{r-1}\}$ such that
 192 for each i , $0 \leq i \leq r-1$, w_{i+1} is $next_M(w_i)$, where $i+1$ is taken modulo r . M/ρ is defined
 193 to be a matching in which each worker not in a pair of ρ stays matched to the same firm
 194 and each worker w_i in ρ is matched to $f_{i+1} = s_M(w_i)$. It can be proven that M/ρ is also a
 195 stable matching. The transformation from M to M/ρ is called the *elimination* of ρ from M .

196 Let $\rho = \{w_0f_0, w_1f_1, \dots, w_{r-1}f_{r-1}\}$ be a rotation. For $0 \leq i \leq r-1$, we say that ρ moves
 197 w_i from f_i to f_{i+1} , and moves f_i from w_i to w_{i+1} . If f is either f_i or is strictly between f_i
 198 and f_{i+1} in w_i 's list, then we say that ρ moves w_i below f . Similarly, ρ moves f_i above w if
 199 w is w_i or between w_i and w_{i+1} in f_i 's list.

200 2.3 The rotation poset

201 A rotation ρ' is said to *precede* another rotation ρ , denoted by $\rho' \prec \rho$, if ρ' is eliminated in
 202 every sequence of eliminations from M_0 to a stable matching in which ρ is exposed. Thus,
 203 the set of rotations forms a partial order via this precedence relationship. The partial order
 204 on rotations is called *rotation poset* and denoted by Π .

205 ▶ **Lemma 3** ([15], Lemma 3.2.1). *For any worker w and firm f , there is at most one rotation
 206 that moves w to f , w below f , or f above w . Moreover, if ρ_1 moves w to f and ρ_2 moves w
 207 from f then $\rho_1 \prec \rho_2$.*

208 ▶ **Lemma 4** ([15], Lemma 3.3.2). *Π contains at most $O(n^2)$ rotations and can be computed
 209 in polynomial time.*

210 ▶ **Definition 5.** *A closed set of a poset is a set S of elements of the poset such that if an
 211 element is in S then all of its predecessors are also in S .*

212 There is a one-to-one relationship between the stable matchings and the closed subsets of Π .
 213 Given a closed set S , the corresponding matching M is found by eliminating the rotations
 214 starting from M_0 according to the topological ordering of the elements in the set S . We say
 215 that S generates M .

216 Let S be a subset of the elements of a poset, and let v be an element in S . We say that v
 217 is a *minimal* element in S if there are no predecessors of v in S . Similarly, v is a *maximal*
 218 element in S if it has no successors in S . The *Hasse diagram* of a poset is a directed graph
 219 with a vertex for each element in the poset, and an edge from x to y if $x \prec y$ and there is no
 220 z such that $x \prec z \prec y$. In other words, all precedences implied by transitivity are suppressed.

221 **2.4 Sublattice and semi-sublattice**

222 A *sublattice* \mathcal{L}' of a distributive lattice \mathcal{L} is subset of \mathcal{L} such that for any two elements
 223 $x, y \in \mathcal{L}$, $x \vee y \in \mathcal{L}'$ and $x \wedge y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$, where \vee and \wedge are the join and
 224 meet operations of lattice \mathcal{L} . A *join semi-sublattice* \mathcal{L}' of a distributive lattice \mathcal{L} is subset of
 225 \mathcal{L} such that for any two elements $x, y \in \mathcal{L}$, $x \vee y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$. Similarly, *meet*
 226 *semi-sublattice* \mathcal{L}' of a distributive lattice \mathcal{L} is subset of \mathcal{L} such that for any two elements
 227 $x, y \in \mathcal{L}$, $x \wedge y \in \mathcal{L}'$ whenever $x, y \in \mathcal{L}'$. Note that \mathcal{L}' is a sublattice of \mathcal{L} iff \mathcal{L}' is both join
 228 and meet semi-sublattice of \mathcal{L} .

229 ► **Proposition 6.** *Let A be an instance of stable matching and let B be another instance
 230 obtained from A by changing the lists of only one side, either workers or firms, but not both.
 231 Then the matchings in $\mathcal{M}_A \cap \mathcal{M}_B$ form a sublattice in each of the two lattices.*

232 ► **Corollary 7.** *Let A be an instance of stable matching and let B_1, \dots, B_k be other instances
 233 obtained from A each by changing the lists of only one side, either workers or firms, but not
 234 both. Then the matchings in $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \dots \cap \mathcal{M}_{B_k}$ form a sublattice in \mathcal{M}_A .*

235 This corollary gives another justification for Algorithm D, motivated by [21]. This
 236 modified Deferred Algorithm works when errors are only on one side. Algorithm D extends
 237 this to errors on both sides however it has exponential runtime.

238 This motivates us to characterize sublattices in the lattice of stable matchings. In
 239 Section 7.1, we show that for any instance B obtained by permuting the preference list of
 240 one worker or one firm, $\mathcal{M}_{A \setminus B}$ forms a semi-sublattice of \mathcal{L}_A (Lemma 31). In particular, if
 241 the list of a worker is permuted, $\mathcal{M}_{A \setminus B}$ forms a join semi-sublattice of \mathcal{L}_A , and if the list of
 242 a firm is permuted, $\mathcal{M}_{A \setminus B}$ forms a meet semi-sublattice of \mathcal{L}_A . In both cases, $\mathcal{M}_A \cap \mathcal{M}_B$ is
 243 a sublattice of \mathcal{L}_A and of \mathcal{L}_B as shown in Proposition 6.

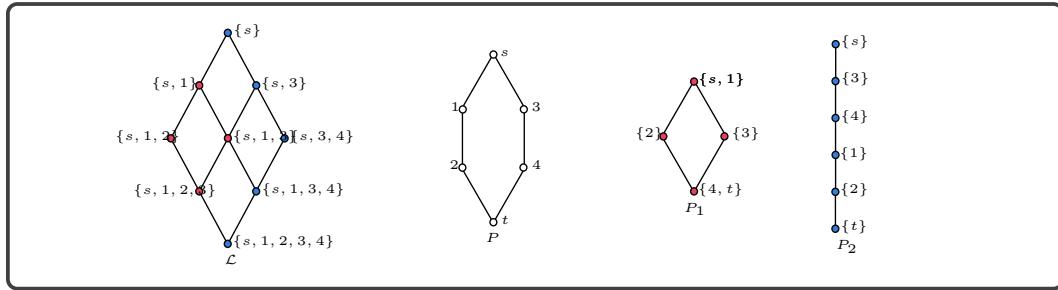
244 **3 Birkhoff's Theorem on Sublattices**

245 Let Π be a finite poset. For simplicity of notation, in this paper we will assume that Π must
 246 have two *dummy elements* s and t ; the remaining elements will be called *proper elements* and
 247 the term *element* will refer to proper as well as dummy elements. The element s precedes all
 248 other elements and t succeeds all other elements in Π . A *proper closed set* of Π is any closed
 249 set that contains s and does not contain t . It is easy to see that the set of all proper closed
 250 sets of Π form a distributive lattice under the operations of set intersection and union. We
 251 will denote this lattice by $L(\Pi)$. The following has also been called *the fundamental theorem*
 252 *for finite distributive lattices*.

253 ► **Theorem 8.** *(Birkhoff [7]) Every finite distributive lattice \mathcal{L} is isomorphic to $L(\Pi)$, for
 254 some finite poset Π .*

255 Our application of Birkhoff's Theorem deals with the sublattices of a finite distributive
 256 lattice. First, in Definition 9 we state the critical operation of *compression of a poset*.

257 ► **Definition 9.** *Given a finite poset Π , first partition its elements; each subset will be called
 258 a *meta-element*. Define the following precedence relations among the meta-elements: if x, y
 259 are elements of Π such that x is in meta-element X , y is in meta-element Y and x precedes
 260 y , then X precedes Y . Assume that these precedence relations yield a partial order, say Q ,
 261 on the meta-elements (if not, this particular partition is not useful for our purpose). Let*



■ **Figure 1** Two examples of compressions. Lattice $\mathcal{L} = L(P)$. P_1 and P_2 are compressions of P , and they generate the sublattices in \mathcal{L} , of red and blue elements, respectively. The black edges are directed from top to bottom so higher elements are predecessors of lower elements.

262 Π' be any partial order on the meta-elements such that the precedence relations of Q are a
 263 subset of the precedence relations of Π' . Then Π' will be called a compression of Π . Let A_s
 264 and A_t denote the meta-elements of Π' containing s and t , respectively.

265 For examples of compressions see Figure 1. Clearly, A_s precedes all other meta-elements
 266 in Π' and A_t succeeds all other meta-elements in Π' . Once again, by a *proper closed set* of
 267 Π' we mean a closed set of Π' that contains A_s and does not contain A_t . Then the lattice
 268 formed by the set of all proper closed sets of Π' will be denoted by $L(\Pi')$.

269 3.1 An alternative view of compression

270 In this section we give an alternative definition of compression of a poset; this will be used
 271 in the rest of the paper. The advantage of this definition is that it is much easier to work
 272 with for the applications presented later. Its drawback is that several different sets of edges
 273 may yield the same compression. Therefore, this definition is not suitable for stating a
 274 one-to-one correspondence between sublattices of \mathcal{L} and compressions of Π . Finally we show
 275 that any compression Π' obtained using the first definition can also be obtained via the
 276 second definition and vice versa (Proposition 10), hence showing that the two definitions are
 277 equivalent for our purposes. See Appendix C for more details.

278 We are given a poset Π for a stable matching instance; let \mathcal{L} be the lattice it generates.
 279 Let $H(\Pi)$ denote the Hasse diagram of Π . Consider the following operations to derive a
 280 new poset Π' : Choose a set E of directed edges to add to $H(\Pi)$ and let H_E be the resulting
 281 graph. Let H' be the graph obtained by shrinking the strongly connected components of
 282 H_E ; each strongly connected component will be called a meta-rotation of Π' as defined
 283 in Definition 9. The edges which are not shrunk will define a DAG, H' , on the strongly
 284 connected components. These edges give precedence relations among meta-rotation for poset
 285 Π' .

286 Let \mathcal{L}' be the sublattice of \mathcal{L} generated by Π' . We will say that the set of edges E defines
 287 \mathcal{L}' . It can be seen that each set E uniquely defines a sublattice $L(\Pi')$; however, there may
 288 be multiple sets that define the same sublattice. See Figure 2 for examples of sets of edges
 289 which define sublattices.

290 ▶ **Proposition 10.** *The two definitions of compression of a poset are equivalent.*

291 For a (directed) edge $e = uv \in E$, u is called the *tail* and v is called the *head* of e . Let I
 292 be a closed set of Π . Then we say that: I separates an edge $uv \in E$ if $v \in I$ and $u \notin I$; I

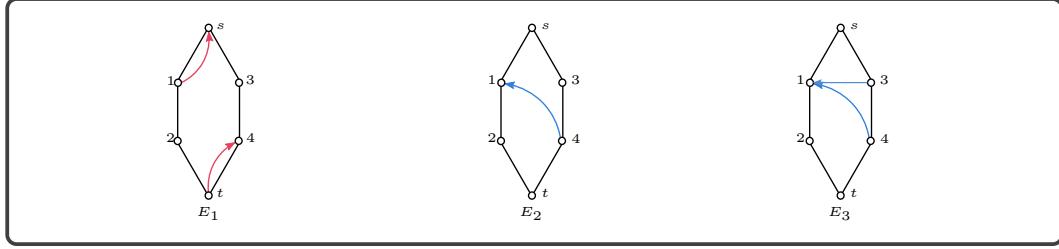


Figure 2 E_1 (red edges) and E_2 (blue edges) define the sublattices in Figure 1, of red and blue elements. E_2 and E_3 define the same compression and represent the same sublattice. All black edges in E_1 , E_2 and E_3 are directed from top to bottom (not shown in the figure).

293 crosses an edge $uv \in E$ if $u \in I$ and $v \notin I$. If I does not separate or cross any edge $uv \in E$,
 294 I is called a *splitting* set w.r.t. E .

295 ► **Lemma 11.** Let \mathcal{L}' be a sublattice of \mathcal{L} and E be a set of edges defining \mathcal{L}' . A matching
 296 M is in \mathcal{L}' iff the closed subset I generating M does not separate any edge $uv \in E$.

297 ► **Remark 12.** We may assume w.l.o.g. that the set E defining \mathcal{L}' is *minimal* in the following
 298 sense: There is no edge $uv \in E$ such that uv is not separated by any closed set of Π . Observe
 299 that if there is such an edge, then $E \setminus \{uv\}$ defines the same sublattice \mathcal{L}' . Similarly, there is
 300 no edge $uv \in E$ such that each closed set separating uv also separates another edge in E .

301 ► **Definition 13.** W.r.t. an element v in a poset Π , we define four useful subsets of Π :
 302 $I_v = \{r \in \Pi : r \prec v\}$, $J_v = \{r \in \Pi : r \preceq v\} = I_v \cup \{v\}$, $I'_v = \{r \in \Pi : r \succ v\}$, $J'_v = \{r \in \Pi : r \succeq v\} = I'_v \cup \{v\}$. Notice that $I_v, J_v, \Pi \setminus I'_v, \Pi \setminus J'_v$ are all closed sets.

304 ► **Lemma 14.** Both J_v and $\Pi \setminus J'_u$ separate uv for each $uv \in E$.

305 **Proof.** Since uv is in E , u cannot be in J_v ; otherwise, there is no closed subset separating
 306 uv , contradicting Remark 12. Hence, J_v separates uv for all uv in E . Similarly, since uv is
 307 in E , v cannot be in J'_u . Therefore, $\Pi \setminus J'_u$ contains v but not u , and thus separates uv . ◀

308 4 Setting I

309 Under Setting I, the given lattice \mathcal{L} has sublattices \mathcal{L}_1 and \mathcal{L}_2 that partition \mathcal{L} . The main
 310 structural fact for this setting is:

311 ► **Theorem 15.** Let \mathcal{L}_1 and \mathcal{L}_2 be sublattices of \mathcal{L} such that \mathcal{L}_1 and \mathcal{L}_2 partition \mathcal{L} . Then
 312 there exist sets of edges E_1 and E_2 defining \mathcal{L}_1 and \mathcal{L}_2 such that they form an alternating
 313 path from t to s .

314 We will prove this theorem in the context of stable matchings. Let E_1 and E_2 be any two
 315 sets of edges defining \mathcal{L}_1 and \mathcal{L}_2 , respectively. We will show that E_1 and E_2 can be adjusted
 316 so that they form an alternating path from t to s , without changing the corresponding
 317 compressions.

318 ► **Lemma 16.** There must exist a path from t to s composed of edges in E_1 and E_2 .

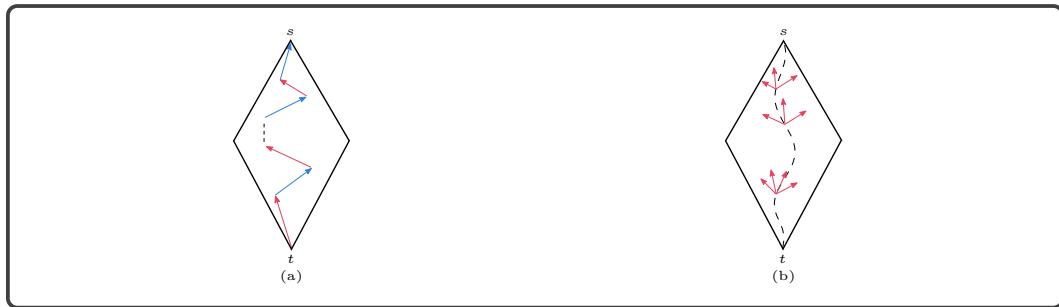


Figure 3 Examples of: (a) canonical path, and (b) bouquet.

319 Let Q be a path from t to s according to Lemma 16. Partition Q into subpaths Q_1, \dots, Q_k
 320 such that each Q_i consists of edges in either E_1 or E_2 and $E(Q_i) \cap E(Q_{i+1}) = \emptyset$ for all
 321 $1 \leq i \leq k-1$. Let r_i be the rotation at the end of Q_i except for $i=0$ where $r_0 = t$.
 322 Specifically, $t = r_0 \rightarrow r_1 \rightarrow \dots \rightarrow r_k = s$ in Q . Lemma 11 can be used to show that each Q_i
 323 can be replaced by a direct edge from r_{i-1} to r_i , and furthermore, all edges not in Q can be
 324 removed.

325 ▶ **Lemma 17.** *Let Q_i consist of edges in E_α ($\alpha = 1$ or 2). Q_i can be replaced by an edge*
 326 *from r_{i-1} to r_i where $r_{i-1}r_i \in E_\alpha$.*

327 ▶ **Lemma 18.** *Edges in $E_1 \cup E_2$ but not in Q can be removed.*

328 By Lemma 17 and Lemma 18, $r_0r_1, \dots, r_{k-2}r_{k-1}, r_{k-1}r_k$ are all edges in E_1 and E_2 and
 329 they alternate between E_1 and E_2 . Therefore, we have Theorem 15. An illustration of such
 330 a path is given in Figure 3(a).

331 ▶ **Proposition 19.** *There exists a sequence of rotations $r_0, r_1, \dots, r_{2k}, r_{2k+1}$ such that a*
 332 *closed subset generates a matching in \mathcal{L}_1 iff it contains r_{2i} but not r_{2i+1} for some $0 \leq i \leq k$.*

333 5 Setting II

334 Under Setting II, the given lattice \mathcal{L} can be partitioned into a sublattice \mathcal{L}_1 and a semi-
 335 sublattice \mathcal{L}_2 . We assume that \mathcal{L}_2 is a join semi-sublattice. Clearly by reversing the order
 336 of \mathcal{L} , the case of meet semi-sublattice is also covered. The next theorem, which generalizes
 337 Theorem 15, gives a sufficient characterization of a set of edges E defining \mathcal{L}_1 .

338 ▶ **Theorem 20.** *There exists a set of edges E defining sublattice \mathcal{L}_1 such that:*

- 339 1. *The set of tails T_E of edges in E forms a chain in Π .*
- 340 2. *There is no path of length two consisting of edges in E .*
- 341 3. *For each $r \in T_E$, let $F_r = \{v \in \Pi : rv \in E\}$. Then any two rotations in F_r are*
 342 *incomparable.*
- 343 4. *For any $r_i, r_j \in T_E$ where $r_i \prec r_j$, there exists a splitting set containing all rotations in*
 344 *$F_{r_i} \cup \{r_i\}$ and no rotations in $F_{r_j} \cup \{r_j\}$.*

345 A set E satisfying Theorem 20 will be called a *bouquet*. For each $r \in T_E$, let $L_r =$
 346 $\{rv \mid v \in F_r\}$. Then L_r will be called a *flower*. Observe that the bouquet E is partitioned
 347 into flowers. These notions are illustrated in Figure 3(b). The black path, directed from s

30:10 A Study of Stable Matching Lattices of "Nearby" Instance

348 to t , is the chain mentioned in Theorem 20 and the red edges constitute E . Observe that
 349 the tails of edges E lie on the chain. For each such tail, the edges of E outgoing from it
 350 constitute a flower.

351 Let E be an arbitrary set of edges defining \mathcal{L}_1 . We will show that E can be modified
 352 so that the conditions in Theorem 20 are satisfied. Let S be a splitting set of Π . In other
 353 words, S is a closed subset such that for all $uv \in E$, either u, v are both in S or u, v are both
 354 in $\Pi \setminus S$. We can now replace paths with single edges as explained below.

355 ▶ **Lemma 21.** *There is a unique maximal rotation in $T_E \cap S$.*

356 Denote by r the unique maximal rotation in $T_E \cap S$. Let $R_r = \{v \in \Pi : \text{there is a path}$
 357 from r to v using edges in $E\}$, $E_r = \{uv \in E : u, v \in R_r\}$, $G_r = \{R_r, E_r\}$. Note that $r \in R_r$.
 358 For each $v \in R_r$ there exists a path from r to v and $r \in S$. Since S does not cross any edge
 359 in the path, v must also be in S . Therefore, $R_r \subseteq S$.

360 ▶ **Lemma 22.** *Let $u \in (T_E \cap S) \setminus R_r$ such that $u \succ x$ for $x \in R_r$. Then we can replace each*
 361 $uv \in E$ with rv .

362 Keep replacing edges according to Lemma 22 until there is no $u \in (T_E \cap S) \setminus R_r$ such
 363 that $u \succ x$ for some $x \in R_r$.

364 ▶ **Lemma 23.** *Let $X = \{v \in S : v \succeq x \text{ for some } x \in R_r\}$. Then: $S \setminus X$ is a closed subset;*
 365 *$S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$; $(S \setminus X) \cap R_r = \emptyset$; $S \setminus X$ is a splitting set.*

366 ▶ **Lemma 24.** *E_r can be replaced by the following set of edges: $E'_r = \{rv : v \in R_r\}$.*

367 **Proof of Theorem 20.** To begin, let $S_1 = \Pi$ and let r_1 be the unique maximal rotation
 368 according to Lemma 21. Then we can replace edges according to Lemma 22 and Lemma 24.
 369 After replacing, r_1 is the only tail vertex in G_{r_1} . By Lemma 23, there exists a set X such
 370 that $S_1 \setminus X$ does not contain any vertex in R_{r_1} and contains all other tail vertices in T_E
 371 except r_1 . Moreover, $S_1 \setminus X$ is a splitting set. Hence, we can set $S_2 = S_1 \setminus X$ and repeat.

372 Let r_1, \dots, r_k be the rotations found in the above process. Since r_i is the unique maximal
 373 rotation in $T_E \cap S_i$ for all $1 \leq i \leq k$ and $S_1 \supset S_2 \supset \dots \supset S_k$, we have $r_1 \succ r_2 \succ \dots \succ r_k$. By
 374 Lemma 24, for each $1 \leq i \leq k$, E_{r_i} consists of edges $r_i v$ for $v \in R_{r_i}$. Therefore, there is no
 375 path of length two composed of edges in E and condition 2 is satisfied. Moreover, r_1, \dots, r_k
 376 are exactly the tail vertices in T_E , which gives condition 1.

377 Let r be a rotation in T_E and consider $u, v \in F_r$. Moreover, assume that $u \prec v$. A closed
 378 subset I separating rv contains v but not r . Since I is a closed subset and $u \prec v$, I contains
 379 u . Therefore, I also separates ru , contradicting the assumption in Remark 12. The same
 380 argument applies when $v \prec u$. Therefore, u and v are incomparable as stated in condition 3.

381 Finally, let $r_i, r_j \in T_E$ where $r_i \prec r_j$. By the construction given above, $S_j \supset S_{j-1} \supset$
 382 $\dots \supset S_i$, $R_{r_j} \subseteq S_j \setminus S_{j-1}$ and $R_{r_i} \subseteq S_i$. Therefore, S_i contains all rotations in R_{r_i} but none
 383 of the rotations in R_{r_j} , giving condition 4 which can be restated as Proposition 25. ◀

384 ▶ **Proposition 25.** *There exists a sequence of rotations $r_1 \prec \dots \prec r_k$ and a set F_{r_i} for each*
 385 *$1 \leq i \leq k$ such that a closed subset generates a matching in \mathcal{L}_1 if and only if whenever it*
 386 *contains a rotation in F_{r_i} , it must also contain r_i .*

FINDBOUQUET(Π):

Input: A poset Π .

Output: A set E of edges defining \mathcal{L}_1 .

1. Initialize: Let $S = \Pi, E = \emptyset$.
2. If M_z is in \mathcal{L}_1 : go to Step 3. Else: $r = t$, go to Step 5.
3. $r = \text{FINDNEXTTAIL}(\Pi, S)$.
4. If r is not NULL: Go to Step 5. Else: Go to Step 7.
5. $F_r = \text{FINDFLOWER}(\Pi, S, r)$.
6. Update:
 - a. For each $u \in F_r$: $E \leftarrow E \cup \{ru\}$.
 - b. $S \leftarrow S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$.
 - c. Go to Step 3.
7. Return E .

Figure 4 Algorithm for finding a bouquet.

6 Algorithm for Finding a Bouquet

In this section, we give an algorithm for finding a bouquet. Let \mathcal{L} be a distributive lattice that can be partitioned into a sublattice \mathcal{L}_1 and a semi-sublattice \mathcal{L}_2 . Then given a poset Π of \mathcal{L} and a membership oracle, which determines if a matching of \mathcal{L} is in \mathcal{L}_1 or not, the algorithm returns a bouquet defining \mathcal{L}_1 .

By Theorem 20, the set of tails T_E forms a chain C in Π . The idea of our algorithm, given in Figure 4, is to find the flowers according to their order in C . Specifically, a splitting set S is maintained such that at any point, all flowers outside of S are found. At the beginning, S is set to Π and becomes smaller as the algorithm proceeds. Step 2 checks if M_z is a matching in \mathcal{L}_1 or not. If $M_z \notin \mathcal{L}_1$, the closed subset $\Pi \setminus \{t\}$ separates an edge in E according to Lemma 11. Hence, the first tail on C must be t . Otherwise, the algorithm jumps to Step 3 to find the first tail. Each time a tail r is found, Step 5 immediately finds the flower L_r corresponding to r . The splitting set S is then updated so that S no longer contains L_r but still contains the flowers that have not been found yet. Next, our algorithm continues to look for the next tail inside the updated S . If no tail is found, it terminates.

► **Lemma 26.** *Let v be a rotation in Π . Let $S \subseteq \Pi$ such that both S and $S \cup \{v\}$ are closed subsets. If S generates a matching in \mathcal{L}_1 and $S \cup \{v\}$ generates a matching in \mathcal{L}_2 , v is the head of an edge in E . If S generates a matching in \mathcal{L}_2 and $S \cup \{v\}$ generates a matching in \mathcal{L}_1 , v is the tail of an edge in E .*

Proof. Suppose that S generates a matching in \mathcal{L}_1 and $S \cup \{v\}$ generates a matching in \mathcal{L}_2 . By Lemma 11, S does not separate any edge in E , and $S \cup \{v\}$ separates an edge $e \in E$. This can only happen if v is the head of e .

A similar argument can be given for the second case. ◀

► **Lemma 27.** *Given a splitting set S , $\text{FINDNEXTTAIL}(\Pi, S)$ (Figure 5) returns the maximal tail vertex in S , or NULL if there is no tail vertex in S .*

30:12 A Study of Stable Matching Lattices of "Nearby" Instance

FINDNEXTTAIL(Π, S):

Input: A poset Π , a splitting set S .

Output: The maximal tail vertex in S , or NULL if there is no tail vertex in S .

1. Compute the set V of rotations v in S such that:

- $\Pi \setminus I'_v$ generates a matching in \mathcal{L}_1 .
- $\Pi \setminus J'_v$ generates a matching in \mathcal{L}_2 .

2. If $V \neq \emptyset$ and there is a unique maximal element v in V : Return v .

Else: Return NULL.

Figure 5 Subroutine for finding the next tail.

FINDFLOWER(Π, S, r):

Input: A poset Π , a tail vertex r and a splitting set S containing r .

Output: The set $F_r = \{v \in \Pi : rv \in E\}$.

1. Compute $X = \{v \in I_r : J_v \text{ generates a matching in } \mathcal{L}_1\}$.

2. Let $Y = \bigcup_{v \in X} J_v$.

3. If $Y = \emptyset$ and $M_0 \in \mathcal{L}_2$: Return $\{s\}$.

4. Compute the set V of rotations v in S such that:

- $Y \cup I_v$ generates a matching in \mathcal{L}_1 .
- $Y \cup J_v$ generates a matching in \mathcal{L}_2 .

5. Return V .

Figure 6 Subroutine for finding a flower.

412 ► **Lemma 28.** Given a tail vertex r and a splitting set S containing r , FINDFLOWER(Π, S, r)
413 (Figure 6) correctly returns F_r .

414 ► **Theorem 29 (h).** FINDBOUQUET(Π), given in Figure 4, returns a set of edges defining
415 \mathcal{L}_1 .

416 **Proof.** From Lemmas 27 and 28, it suffices to show that S is updated correctly in Step 6(b).
417 To be precise, we need that

$$418 \quad S \setminus \bigcup_{u \in F_r \cup \{r\}} J'_u$$

419 must still be a splitting set, and contains all flowers that have not been found. This follows
420 from Lemma 23 by noticing that

$$421 \quad \bigcup_{u \in F_r \cup \{r\}} J'_u = \{v \in \Pi : v \succeq u \text{ for some } u \in R_r\}.$$

422

423 Clearly, a sublattice of \mathcal{L} must also be a semi-sublattice. Therefore, FINDBOUQUET can
424 be used to find a canonical path described in Section 4. The same algorithm can be used to
425 check if $M_A \cap M_B = \emptyset$. Let E be the edge set given by the FINDBOUQUET algorithm and
426 H_E be the corresponding graph obtained by adding E to the Hasse diagram of the original

$\begin{array}{c cccc} 1 & b & a & c & d \\ 2 & a & b & c & d \\ 3 & d & c & a & b \\ 4 & c & d & a & b \end{array}$	$\begin{array}{c cccc} 1 & \color{red}{c} & a & \color{red}{b} & d \\ 2 & a & b & c & d \\ 3 & d & c & a & b \\ 4 & c & d & a & b \end{array}$	$\begin{array}{c cccc} a & 1 & 2 & 3 & 4 \\ b & 2 & 1 & 3 & 4 \\ c & 3 & 1 & 4 & 2 \\ d & 4 & 3 & 1 & 2 \end{array}$
firms' preferences in A	firms' preferences in B	workers' preferences in both instances

■ **Figure 7** An example in which $\mathcal{M}_{A \setminus B}$ is not a sublattice of \mathcal{L}_A .

427 rotation poset Π of \mathcal{L}_A . If H_E has a single strongly connected component, the compression
 428 Π' has a single meta-element and represents the empty lattice.

429 7 Finding a Fully Robust Stable Matching

430 Consider the setting given in the Introduction, with S being the domain of errors, one of
 431 which is introduced in instance A . We show how to use the algorithm in Section 6 to find
 432 the poset generating all fully robust matchings w.r.t. S . We then show how this poset can
 433 yield a fully robust matching that maximizes, or minimizes, a given weight function.

434 7.1 Studying semi-sublattices is necessary and sufficient

435 Let A be a stable matching instance, and B be an instance obtained by permuting the
 436 preference list of one worker or one firm. Lemma 30 gives an example of a permutation so
 437 that $\mathcal{M}_{A \setminus B}$ is not a sublattice of \mathcal{L}_A , hence showing that the case studied in Section 4 does
 438 not suffice to solve the problem at hand. On the other hand, for all such instances B , Lemma
 439 31 shows that $\mathcal{M}_{A \setminus B}$ forms a semi-sublattice of \mathcal{L}_A and hence the case studied in Section 5
 440 does suffice.

441 The next lemma pertains to the example given in Figure 7, in which the set of workers is
 442 $B = \{a, b, c, d\}$ and the set of firms is $G = \{1, 2, 3, 4\}$. Instance B is obtained from instance
 443 A by permuting firm 1's list.

444 ► **Lemma 30.** *There exist stable matching instances A and B differing by one agent's
 445 preference list such that $\mathcal{M}_{A \setminus B}$ is not a sublattice of \mathcal{L}_A .*

446 ► **Lemma 31.** *For any instance B obtained by permuting the preference list of one worker
 447 or one firm, $\mathcal{M}_{A \setminus B}$ forms a semi-sublattice of \mathcal{L}_A .*

448 ► **Proposition 32.** *A set of edges defining the sublattice \mathcal{L}' , consisting of matchings in
 449 $\mathcal{M}_A \cap \mathcal{M}_B$, can be computed in polynomial time.*

450 7.2 Proof of Theorem 2

451 In this section, we will prove Theorem 2 as well as a slight extension; the latter uses ideas
 452 from [18]. Let B_1, \dots, B_k be polynomially many instances in the domain $D \subset T$, as defined
 453 in the Introduction. Let E_i be the set of edges defining $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for all $1 \leq i \leq k$. By
 454 Corollary 7, $\mathcal{L}' = \mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \dots \cap \mathcal{M}_{B_k}$ is a sublattice of \mathcal{L}_A .

30:14 A Study of Stable Matching Lattices of "Nearby" Instance

455 ► **Lemma 33.** $E = \bigcup_i E_i$ defines \mathcal{L}' .

456 **Proof.** By Lemma 11, it suffices to show that for any closed subset I , I does not separate
457 an edge in E iff I generates a matching in \mathcal{L}' .

458 I does not separate an edge in E iff I does not separate any edge in E_i for all $1 \leq i \leq k$
459 iff the matching generated by I is in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for all $1 \leq i \leq k$ by Lemma 11. ◀

460 By Lemma 33, a compression Π' generating \mathcal{L}' can be constructed from E as described in
461 Section 3.1. By Proposition 32, we can compute each E_i , and hence, Π' in polynomial time.
462 Clearly, Π' can be used to check if a fully robust stable matching exists. To be precise, a
463 fully robust stable matching exists iff there exists a proper closed subset of Π' . This happens
464 iff s and t belong to different meta-rotations in Π' , an easy to check condition. Hence, we
465 have Theorem 2.

466 7.3 Finding maximum weight fully robust stable matchings

467 We can use Π' to obtain a fully robust stable matching M maximizing $\sum_{wf \in M} W_{wf}$ by
468 applying the algorithm of [19]. Specifically, let $H(\Pi')$ be the Hasse diagram of Π' . Then each
469 pair wf for $w \in \mathcal{W}$ and $f \in \mathcal{F}$ can be associated with two vertices u_{wf} and v_{wf} in $H(\Pi')$ as
470 follows:

- 471 ■ If there is a rotation r moving w to f , u_{wf} is the meta-rotation containing r . Otherwise,
472 u_{wf} is the meta-rotation containing s .
- 473 ■ If there is a rotation r moving w from f , v_{wf} is the meta-rotation containing r . Otherwise,
474 v_{wf} is the meta-rotation containing t .

475 By Lemma 3 and the definition of compression, $u_{wf} \prec v_{wf}$. Hence, there is a path
476 from u_{wf} to v_{wf} in $H(\Pi')$. We can then add weights to edges in $H(\Pi')$, as stated in [19].
477 Specifically, we start with weight 0 on all edges and increase weights of edges in a path from
478 u_{wf} to v_{wf} by w_{wf} for all pairs wf . A fully robust stable matching maximizing $\sum_{wf \in M} W_{bwf}$
479 can be obtained by finding a maximum weight ideal cut in the constructed graph. An efficient
480 algorithm for the latter problem is given in [19].

481 8 Discussion

482 The primary focus of this paper is the study of "nearby" stable matching instances where
483 a single agent permutes their preference list. A number of new questions arise: give a
484 polynomial time algorithm for the problem mentioned in the Introduction, of finding a robust
485 stable matching as defined in [19] — given a probability distribution on the domain of errors
486 — even when the error is an arbitrary permutation; and extend to the stable roommate
487 problem and incomplete preference lists [15, 20], as well as popular matchings [10, 16].

488 Next, we give a hypothetical setting to show potential application of our work to the
489 issue of incentive compatibility. Let A be an instance of stable matching over n workers
490 and n firms. Assume that all $2n$ agents have a means of making their preference lists public
491 simultaneously and a dominant firm, say f , is given the task of computing and announcing
492 a stable matching. Once the matching is announced, all agents can verify that it is indeed
493 stable. It turns out that firm f can cheat and improve its match as follows: f changes
494 its preference list to obtain instance B which is identical to A for all other agents, and
495 computes a matching that is stable for A as well as B using Theorem 2. The other agents
496 will be satisfied that this matching is indeed stable for instance A and f 's cheating may go
497 undetected.

498 Finally, considering the number of new and interesting matching markets being defined
 499 on the Internet, e.g., see [13], it will not be surprising if new, deeper structural facts about
 500 stable matching lattices find suitable applications. For this reason, the problem initiated
 501 in [18], which appears to be a fundamental one, deserves further work. In particular, we
 502 leave the question of extending our work to the case when the two instances A and B are not
 503 nearby but arbitrary, i.e., when multiple agents simultaneously change their preference lists.

504 ————— **References** —————

- 505 1 Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. Strategy-proofness versus efficiency
 506 in matching with indifferences: Redesigning the NYC high school match. *American Economic
 507 Review*, 99(5):1954–78, 2009.
- 508 2 Atila Abdulkadiroğlu, Parag A Pathak, Alvin E Roth, and Tayfun Sönmez. The boston public
 509 school match. *American Economic Review*, 95(2):368–371, 2005.
- 510 3 Atila Abdulkadiroğlu and Tayfun Sonmez. School choice: A mechanism design approach.
 511 *American economic review*, 93(3):729–747, 2003.
- 512 4 H. Aziz, P. Biro, T. Fleiner, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching
 513 with uncertain pairwise preferences. In *International Joint Conference on Autonomous
 514 Agents and Multiagent Systems*, pages 344–352, 2017.
- 515 5 H. Aziz, P. Biro, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable matching with
 516 uncertain linear preferences. In *International Symposium on Algorithmic Game Theory*, pages
 517 195–206, 2016.
- 518 6 A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in
 519 Applied Mathematics. Princeton University Press, October 2009.
- 520 7 Garrett Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.
- 521 8 Charles Blair. Every finite distributive lattice is a set of stable matchings. *Journal of
 522 Combinatorial Theory, Series A*, 37(3):353–356, 1984.
- 523 9 G. C. Calafiore and L. El Ghaoui. On distributionally robust chance-constrained linear
 524 programs. *Jour. of Optimization Theory and Applications*, 130(1), December 2006.
- 525 10 Agnes Cseh, Yuri Faenza, Telikepalli Kavitha, and Vladlena Powers. Understanding popular
 526 matchings via stable matchings. *SIAM Journal on Discrete Mathematics*, 36(1):188–213, 2022.
- 527 11 L. E. Dubins and D. A. Freedman. Machiavelli and the gale-shapley algorithm. *The American
 528 Mathematical Monthly*, 88(7):485–494, 1981.
- 529 12 Federico Echenique, Nicole Immorlica, and Vijay V. Vazirani, editors. *Online and Matching-
 530 Based Market Design*. Cambridge University Press, to appear. <https://www.ics.uci.edu/~vazirani/Chapter1.pdf>.
- 532 13 Simons Institute for the Theory of Computing. Online and matching-based market design,
 533 2019. <https://simons.berkeley.edu/programs/market2019>.
- 534 14 David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The
 535 American Mathematical Monthly*, 69(1):9–15, 1962.
- 536 15 Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*.
 537 MIT press, 1989.
- 538 16 Telikepalli Kavitha. Matchings, critical nodes, and popular solutions. In *41st IARCS Annual
 539 Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS
 540 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 541 17 Donald Ervin Knuth. *Stable marriage and its relation to other combinatorial problems: An
 542 introduction to the mathematical analysis of algorithms*. American Mathematical Soc., 1997.
- 543 18 Tung Mai and Vijay V. Vazirani. Finding stable matchings that are robust to errors in the
 544 input. In *European Symposium on Algorithms*, 2018.
- 545 19 Tung Mai and Vijay V. Vazirani. A natural generalization of stable matching solved via new
 546 insights into ideal cuts. In arXiv, 2018.
- 547 20 David Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

30:16 A Study of Stable Matching Lattices of "Nearby" Instance

548 21 Shuichi Miyazaki and Kazuya Okamoto. Jointly stable matchings. *Journal of Combinatorial*
549 *Optimization*, 38(2):646–665, 2019.
550 22 Alvin E Roth and Marilda Sotomayor. Two-sided matching. *Handbook of game theory with*
551 *economic applications*, 1:485–541, 1992.
552 23 Richard Stanley. Enumerative combinatorics, vol. 1, Wadsworth and Brooks/Cole, Pacific
553 Grove, CA, 1986; second printing, 1996.

554 A Related Work

555 A.1 Related work

556 The two topics, of stable matching and the design of algorithms that produce solutions that
557 are robust to errors, have been studied extensively for decades and there are today several
558 books on each of them, e.g., see [17, 15, 20] and [9, 6]. Yet, there is a paucity of results at
559 the intersection of these two topics. Indeed, before the publication of [18], we are aware of
560 only two previous works [5, 4]. We remark that the notion of robustness studied in [18] was
561 quite different from that of the previous two works as detailed below.

562 Aziz et al. [5] considered the problem of finding stable matching under uncertain linear
563 preferences. They proposed three different uncertainty models:

- 564 1. Lottery Model: Each agent has a probability distribution over strict preference lists,
565 independent of other agents.
- 566 2. Compact Indifference Model: Each agent has a single weak preference list in which ties
567 may exist. All linear order extensions of this weak order have equal probability.
- 568 3. Joint Probability Model: A probability distribution over preference profiles is specified.
569 They showed that finding the matching with highest probability of being stable is NP-hard
570 for the Compact Indifference Model and the Joint Probability Model. For the very special
571 case that preference lists of one side are certain and the number of uncertain agents of the
572 other side are bounded by a constant, they gave a polynomial time algorithm that works for
573 all three models.

574 The joint probability model is the most powerful and closest to our setting. The main
575 difference is that in their model, there is no base instance, which is called A in our model.
576 The opportunity of finding new structural results arises from our model precisely because we
577 need to consider two “nearby” instances, namely A and B as described above.

578 Aziz et al. [4] introduced a pairwise probability model in which each agent gives the
579 probability of preferring one agent over another for all possible pairs. They showed that the
580 problem of finding a matching with highest probability of being stable is NP-hard even when
581 no agent has a cycle in its certain preferences (i.e., the ones that hold with probability 1).

582 B Proof of Birkhoff’s Theorem using Stable Matching Lattices

583 Omitted proofs can be found in the *Arxiv version*.

584 C Other Omitted Proofs

585 **Proof of Lemma 6.** It suffices to show that $\mathcal{M}_A \cap \mathcal{M}_B$ is a sublattice of \mathcal{L}_A . Assume
586 $|\mathcal{M}_A \cap \mathcal{M}_B| > 1$ and let M_1 and M_2 be two different matchings in $\mathcal{M}_A \cap \mathcal{M}_B$. Let \vee_A and
587 \vee_B be the join operations under A and B respectively. Likewise, let \wedge_A and \wedge_B be the meet
588 operations under A and B .

589 By definition of join operation in Section 2.1, $M_1 \vee_A M_2$ is the matching obtained by
 590 assigning each worker to its less preferred partner (or equivalently, each firm to its more
 591 preferred partner) from M_1 and M_2 according to instance A . Without loss of generality,
 592 assume that B is an instance obtained from A by changing the lists of only firms. Since
 593 the list of each worker is identical in A and B , its less preferred partner from M_1 and M_2 is
 594 also the same in A and B . Therefore, $M_1 \vee_A M_2 = M_1 \vee_B M_2$. A similar argument can be
 595 applied to show that $M_1 \wedge_A M_2 = M_1 \wedge_B M_2$.

596 Hence, $M_1 \vee_A M_2$ and $M_1 \wedge_A M_2$ are both in $\mathcal{M}_A \cap \mathcal{M}_B$ as desired. \blacktriangleleft

597 **Proof of Corollary 7.** Assume $|\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \dots \cap \mathcal{M}_{B_k}| > 1$ and let M_1 and M_2 be two
 598 different matchings in $\mathcal{M}_A \cap \mathcal{M}_{B_1} \cap \dots \cap \mathcal{M}_{B_k}$. Therefore, M_1 and M_2 are in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$
 599 for each $1 \leq i \leq k$. By Proposition 6, $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ is a sublattice of \mathcal{L}_A . Hence, $M_1 \vee_A M_2$
 600 and $M_1 \wedge_A M_2$ are in $\mathcal{M}_A \cap \mathcal{M}_{B_i}$ for each $1 \leq i \leq k$. The claim then follows. \blacktriangleleft

601 **Proof of Proposition 10.** Let Π' be a compression of Π obtained using the first definition.
 602 Clearly, for each meta-rotation in Π' , we can add edges to Π so the strongly connected
 603 component created is precisely this meta-rotation. Any additional precedence relations
 604 introduced among incomparable meta-rotations can also be introduced by adding appropriate
 605 edges.

606 The other direction is even simpler, since each strongly connected component can be
 607 defined to be a meta-rotation and extra edges added can also be simulated by introducing
 608 new precedence constraints. \blacktriangleleft

609 **Proof of Lemma 11.** Let Π' be a compression corresponding to \mathcal{L}' . By Theorem 1, the
 610 matchings in \mathcal{L}' are generated by eliminating rotations in closed subsets of Π' .

611 First, assume I separates $uv \in E$. Moreover, assume $M \in \mathcal{L}'$ for the sake of contradiction,
 612 and let I' be the closed subset of Π' corresponding to M . Let U and V be the meta-rotations
 613 containing u and v respectively. Notice that the sets of rotations in I and I' are identical.
 614 Therefore, $V \in I'$ and $U \notin I'$. Since $uv \in E$, there is an edge from U to V in H' . Hence, I'
 615 is not a closed subset of Π' .

616 Next, assume that I does not separate any $uv \in E$. We show that the rotations in I can
 617 be partitioned into meta-rotations in a closed subset I' of Π' . If I cannot be partitioned
 618 into meta-rotations, there must exist a meta-rotation A such that $A \cap I$ is a non-empty
 619 proper subset of A . Since A consists of rotations in a strongly connected component of H_E ,
 620 there must be an edge uv from $A \setminus I$ to $A \cap I$ in H_E . Hence, I separates uv . Since I is a
 621 closed subset, uv can not be an edge in H . Therefore, $uv \in E$, which is a contradiction. It
 622 remains to show that the set of meta-rotations partitioning I is a closed subset of Π' . Assume
 623 otherwise, there exist meta-rotation $U \in I'$ and $V \notin I'$ such that there exists an edge from U
 624 to V in H' . Therefore, there exists $u \in U$, $v \in V$ and $uv \in E$, which is a contradiction. \blacktriangleleft

625 **Proof of Lemma 16.** Let R denote the set of vertices reachable from t by a path of edges
 626 in E_1 and E_2 . Assume by contradiction that R does not contain s . Consider the matching
 627 M generated by rotations in $\Pi \setminus R$. Without loss of generality, assume that $M \in \mathcal{L}_1$. By
 628 Lemma 11, $\Pi \setminus R$ separates an edge $uv \in E_2$. Therefore, $u \in R$ and $v \in \Pi \setminus R$. Since $uv \in E_2$,
 629 v is also reachable from t by a path of edges in E_1 and E_2 . \blacktriangleleft

630 **Proof of Lemma 17.** A closed subset separating $r_{i-1}r_i$ must separate an edge in Q_i . Moreover,
 631 any closed subset must separate exactly one of $r_0r_1, \dots, r_{k-2}r_{k-1}, r_{k-1}r_k$. Therefore, the set
 632 of closed subsets separating an edge in E_1 (or E_2) remains unchanged. \blacktriangleleft

30:18 A Study of Stable Matching Lattices of "Nearby" Instance

633 **Proof of Lemma 18.** Let e be an edge in $E_1 \cup E_2$ but not in Q . Suppose that $e \in E_1$. Let
634 I be a closed subset separating e . By Lemma 11, the matching generated by I belongs to
635 \mathcal{L}_2 . Since e is not in Q and Q is a path from t to s , I must separate another edge e' in Q .
636 By Lemma 11, I can not separate edges in both E_1 and E_2 . Therefore, e' must also be in
637 E_1 . Hence, the matching generated by I will still be in \mathcal{L}_2 after removing e from E_1 . The
638 argument applies to all closed subsets separating e . \blacktriangleleft

639 **Proof of Lemma 21.** Suppose there are at least two maximal rotations u_1, u_2, \dots, u_k ($k \geq 2$)
640 in $T_E \cap S$. Let v_1, \dots, v_k be the heads of edges containing u_1, u_2, \dots, u_k . For each $1 \leq i \leq k$,
641 let $S_i = J_{u_i} \cup J_{v_j}$ where j is any index such that $j \neq i$. Since u_i and u_j are incomparable,
642 $u_j \notin J_{u_i}$. Moreover, $u_j \notin J_{v_j}$ by Lemma 14. Therefore, $u_j \notin S_i$. It follows that S_i contains
643 u_i and separates $u_j v_j$. Since S_i separates $u_j v_j \in E$, the matching generated by S_i is in \mathcal{L}_2
644 according to Lemma 11.

645 Since $\bigcup_{i=1}^k S_i$ contains all maximal rotations in $T_E \cap S$ and S does not separate any edge
646 in E , $\bigcup_{i=1}^k S_i$ does not separate any edge in E either. Therefore, the matching generated
647 by $\bigcup_{i=1}^k S_i$ is in \mathcal{L}_1 , and hence not in \mathcal{L}_2 . This contradicts the fact that \mathcal{L}_2 is a join
648 semi-sublattice. \blacktriangleleft

649 **Proof of Lemma 22.** We will show that the set of closed subsets separating an edge in E
650 remains unchanged.

651 Let I be a closed subset separating uv . Then I must also separate rv since $r \succ v$.

652 Now suppose I is a closed subset separating rv . We consider two cases:

- 653 ■ If $u \in I$, I must contain x since $u \succ x$. Hence, I separates an edge in the path from r to
654 x .
- 655 ■ If $u \notin I$, I separates uv .

656 \blacktriangleleft

657 **Proof of Lemma 23.** The lemma follows from the claims given below:

658 \triangleright Claim 34. $S \setminus X$ is a closed subset.

659 **Proof.** Let v be a rotation in $S \setminus X$ and u be a predecessor of v . Since S is a closed subset,
660 $u \in S$. Notice that if a rotation is in X , all of its successor must be included. Hence, since
661 $v \notin X$, $u \notin X$. Therefore, $u \in S \setminus X$. \blacktriangleleft

662 \triangleright Claim 35. $S \setminus X$ contains u for each $u \in (T_E \cap S) \setminus R_r$.

663 **Proof.** After replacing edges according to Lemma 22, for each $u \in (T_E \cap S) \setminus R_r$ we must
664 have that u does not succeed any $x \in R_r$. Therefore, $u \notin X$ by the definition of X . \blacktriangleleft

665 \triangleright Claim 36. $(S \setminus X) \cap R_r = \emptyset$.

666 **Proof.** Since $R_r \subseteq X$, $(S \setminus X) \cap R_r = \emptyset$. \blacktriangleleft

667 \triangleright Claim 37. $S \setminus X$ does not separate any edge in E .

668 **Proof.** Suppose $S \setminus X$ separates $uv \in E$. Then $u \in X$ and $v \in S \setminus X$. By Claim 2, u can
669 not be a tail vertex, which is a contradiction. \blacktriangleleft

670 \triangleright Claim 38. $S \setminus X$ does not cross any edge in E .

671 **Proof.** Suppose $S \setminus X$ crosses $uv \in E$. Then $u \in S \setminus X$ and $v \in X$. Let J be a closed subset
672 separating uv . Then $v \in J$ and $u \notin J$.

673 Since $uv \in E$ and $u \in S$, $u \in T_E \cap S$. Therefore, $r \succ u$ by Lemma 21. Since J is a closed
674 subset, $r \notin J$.

675 Since $v \in X$, $v \succeq x$ for $x \in R_r$. Again, as J is a closed subset, $x \in J$.

676 Therefore, J separates an edge in the path from r to x in G_r . Hence, all closed subsets
677 separating uv must also separate another edge in E_r . This contradicts the assumption made
678 in Remark 12. \blacktriangleleft

679 \blacktriangleleft

680 **Proof of Lemma 24.** We will show that the set of closed subsets separating an edge in E_r
681 and the set of closed subset separating an edge in E'_r are identical.

682 Consider a closed subset I separating an edge in $rv \in E'_r$. Since $v \in R_r$, I must separate
683 an edge in E in a path from r to v . By definition, that edge is in E_r .

684 Now let I be a closed subset separating an edge in $uv \in E_r$. Since $uv \in E$, $u \in T_E \cap S$.
685 By Lemma 21, $r \succ u$. Thus, I must also separate $rv \in E'_r$. \blacktriangleleft

686 **Proof of Lemma 27.** Let r be the maximal tail vertex in S .

687 First we show that $r \in V$. By Theorem 20, the set of tails of edges in E forms a chain in
688 Π . Therefore $\Pi \setminus I'_r$ contains all tails in S . Hence, $\Pi \setminus I'_r$ does not separate any edge whose
689 tails are in S . Since S is a splitting set, $\Pi \setminus I'_r$ does not separate any edge whose tails are in
690 $\Pi \setminus S$. Therefore, by Lemma 11, $\Pi \setminus I'_r$ generates a matching in \mathcal{L}_1 . By Lemma 14, $\Pi \setminus J'_r$
691 must separate an edge in E , and hence generates a matching in \mathcal{L}_2 according to Lemma 11.

692 By Lemma 26, any rotation in V must be the tail of an edge in E . Hence, they are all
693 predecessors of r according to Theorem 20. \blacktriangleleft

694 **Proof of Lemma 28.** First we give two crucial properties of the set Y . By Theorem 20, the
695 set of tails of edges in E forms a chain C in Π .

696 \triangleright **Claim 39.** Y contains all predecessors of r in C .

697 **Proof.** Assume that there is at least one predecessor of r in C , and denote by r' the direct
698 predecessor. It suffices to show that $r' \in Y$. By Theorem 20, there exists a splitting set I
699 such that $R_{r'} \subseteq I$ and $R_r \cap I = \emptyset$. Let v be the maximal element in $C \cap I$. Then v is a
700 successor of all tail vertices in I . It follows that J_v does not separate any edges in E inside I .
701 Therefore, $v \in X$. Since $J_v \subseteq Y$, Y contains all predecessors of r in C . \blacktriangleleft

702 \triangleright **Claim 40.** Y does not contain any rotation in F_r .

703 **Proof.** Since Y is the union of closed subset generating matching in \mathcal{L}_1 , Y also generates a
704 matching in \mathcal{L}_1 . By Lemma 11, Y does not separate any edge in E . Since $r \notin Y$, Y must
705 not contain any rotation in F_r . \blacktriangleleft

706 By Claim 1, if $Y = \emptyset$, r is the last tail found in C . Hence, if $M_0 \in \mathcal{L}_2$, s must be in F_r .
707 By Theorem 20, the heads in F_r are incomparable. Therefore, s is the only rotation in C .
708 FINDFLOWER correctly returns $\{s\}$ in Step 3. Suppose such a situation does not happen, we
709 will show that the returned set is F_r .

710 \triangleright **Claim 41.** $V = F_r$.

30:20 A Study of Stable Matching Lattices of "Nearby" Instance

711 **Proof.** Let v be a rotation in V . By Lemma 26, v is a head of some edge e in E . Since Y
712 contains all predecessors of r in C , the tail of e must be r . Hence, $v \in F_r$.

713 Let v be a rotation in F_r . Since Y contains all predecessors of r in C , $Y \cup I_v$ can not
714 separate any edge whose tails are predecessors of r . Moreover, by Theorem 20, the heads in
715 F_r are incomparable. Therefore, I_v does not contain any rotation in F_r . Since Y does not
716 contain any rotation in F_r by the above claim, $Y \cup I_v$ does not separate any edge in E . It
717 follows that $Y \cup I_v$ generates a matching in \mathcal{L}_1 . Finally, $Y \cup J_v$ separates rv clearly, and
718 hence generates a matching in \mathcal{L}_2 . Therefore, $v \in V$ as desired. \blacktriangleleft

719 \blacktriangleleft

720 **Proof of Lemma 30.** $M_1 = \{1a, 2b, 3d, 4c\}$ and $M_2 = \{1b, 2a, 3c, 4d\}$ are stable matching
721 with respect to instance A . Clearly, $M_1 \wedge_A M_2 = \{1a, 2b, 3c, 4d\}$ is also a stable matching
722 under A .

723 In going from A to B , the positions of workers b and c are swapped in firm 1's list. Under
724 B , $1c$ is a blocking pair for M_1 and $1a$ is a blocking pair for M_2 . Hence, M_1 and M_2 are
725 both in $\mathcal{M}_{A \setminus B}$. However, $M_1 \wedge_A M_2$ is a stable matching under B , and therefore is it not in
726 $\mathcal{M}_{A \setminus B}$. Hence, $\mathcal{M}_{A \setminus B}$ is not closed under the \wedge_A operation. \blacktriangleleft

727 **Proof of Lemma 31.** Assume that the preference list of a firm f is permuted. We will show
728 that $\mathcal{M}_{A \setminus B}$ is a join semi-sublattice of \mathcal{L}_A . By switching the role of workers and firms,
729 permuting the list of a worker will result in $\mathcal{M}_{A \setminus B}$ being a meet semi-sublattice of \mathcal{L}_A .

730 Let M_1 and M_2 be two matchings in $\mathcal{M}_{A \setminus B}$. Hence, neither of them are in \mathcal{M}_B . In other
731 words, each has a blocking pair under instance B .

732 Let w be the partner of f in $M_1 \vee_A M_2$. Then w must also be matched to f in either M_1
733 or M_2 (or both). We may assume that w is matched to f in M_1 .

734 Let xy be a blocking pair of M_1 under B . We will show that xy must also be a blocking
735 pair of $M_1 \vee_A M_2$ under B . To begin, the firm y must be f since other preference lists remain
736 unchanged. Since xf is a blocking pair of M_1 under B , $x >_f^B w$. Similarly, $f >_x f'$ where f'
737 is the M_1 -partner of x . Let f'' be the partner of x in $M_1 \vee_A M_2$. Then $f' \geq_x f''$. It follows
738 that $f >_x f''$. Since $x >_f^B w$ and $f >_x f''$, xf must be a blocking pair of $M_1 \vee_A M_2$ under
739 B . \blacktriangleleft

740 **Proof of Proposition 32.** We have that \mathcal{L}' and $\mathcal{M}_{A \setminus B}$ partition \mathcal{L}_A , with $\mathcal{M}_{A \setminus B}$ being a
741 semi-sublattice of \mathcal{L}_A , by Lemma 31. Therefore, FINDBOUQUET(Π) finds a set of edges
742 defining \mathcal{L}' by Theorem 29.

743 By Lemma 4, the input Π to FINDBOUQUET can be computed in polynomial time. Clearly,
744 a membership oracle checking if a matching is in \mathcal{L}' or not can also be implemented efficiently.
745 Since Π has $O(n^2)$ vertices (Lemma 4), any step of FINDBOUQUET takes polynomial time. \blacktriangleleft

746 D Modified Deferred Acceptance Algorithms

747 Omitted algorithms and proofs can be found in the *Arxiv version*.