Reasoning about Optimal Stable Matchings under Partial Information

BAHARAK RASTEGARI, University of Glasgow
ANNE CONDON, University of British Columbia
NICOLE IMMORLICA, Microsoft
ROBERT IRVING, University of Glasgow
KEVIN LEYTON-BROWN, University of British Columbia

We study two-sided matching markets in which participants are initially endowed with partial preference orderings, lacking precise information about their true, strictly ordered list of preferences. We wish to reason about matchings that are stable with respect to agents' true preferences, and which are furthermore optimal for one given side of the market. We present three main results. First, one can decide in polynomial time whether there exists a matching that is stable and optimal under all strict preference orders that refine the given partial orders, and can construct this matching in polynomial time if it does exist. We show, however, that deciding whether a given pair of agents are matched in all or no such optimal stable matchings is co-NP-complete, even under quite severe restrictions on preferences. Finally, we describe a polynomial-time algorithm that decides, given a matching that is stable under the partial preference orderings, whether that matching is stable and optimal for one side of the market under some refinement of the partial orders.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics

Additional Key Words and Phrases: Matching; Market design; Partial information; Optimal stable matching

1. INTRODUCTION

Two-sided matching markets model many practical settings, such as corporate hiring, marriage, and university admission. In such markets, participants are partitioned into two disjoint sets, such as employers and applicants in a job market. Each participant wishes to be matched to a candidate from the other side of the market, and has preferences over potential matches. Stability is perhaps the most desirable and widely studied solution concept in two-sided matching markets. A matching is called stable if no pair of participants would prefer to leave their assigned partners to pair with each other. A stable matching is optimal for one side of the market if there is no stable matching that is preferred by at least one agent on that side of the market. In their seminal work, Gale and Shapley [?] introduced an efficient algorithm for identifying such optimal stable matchings. A rich literature has developed since. See the books by Knuth [?], Gusfield and Irving [?], Roth and Sotomayor [?], and Manlove [?] for excellent introductions and surveys.

A key assumption in much of this literature is that all market participants know their full (and, it is often assumed, strict) preference orderings. This assumption is reasonable in some settings; however, as markets grow large it quickly becomes impractical for participants to assess their precise preference rankings. In this work we focus on two-sided matching markets in which agents are endowed with known, partially ordered preferences which are consistent with unknown, strict preferences [?]. For example, a recent Ph.D. graduate applying to different schools for a faculty position cannot know his

---

This work was supported through a Discovery grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada, and also by grants EP/K01000X/1 and EP/K010042/1 from the Engineering and Physical Sciences Research Council. The work was partly carried out while the first author was at the University of British Columbia.
Author's addresses: B. Rastegari and R. Irving, School of Computing Science, University of Glasgow; email: {baharak.rastegari,rob.irving}@glasgow.ac.uk; A. Condon and K. Leyton-Brown, Department of Computer Science, University of British Columbia; email: {condon,kevinlb}@cs.ubc.ca; N. Immorlica, Microsoft; email: nicimm@gmail.com.

EC'14, June 8–12, 2014, Stanford, CA, USA.
http://dx.doi.org/10.1145/2600057.2602884

or her true preferences over all possible available positions. Instead s/he may know which schools are his or her top-tier choices, which are his or her second-tier choices, and so on.

We seek a stable matching that is optimal for one side of the market, say the employers (schools), with respect to the agents' underlying true, albeit unknown, strict preferences. Finding such a matching is important for two reasons. Firstly, a social planner may want to optimize the solution for one side of the market. For example in the hospitals-residents problem, where junior doctors are assigned to hospitals, a central matching program is usually instructed to pick the stable matching most preferred by the junior doctors. This is precisely what NRMP in US (see, e.g., **?**) and SFAS in Scotland (see, e.g., **?**) do. Secondly, picking an optimal stable matching implies dominant-strategy truthfulness for one side of the market. Note that no mechanism exists that is dominant-strategy truthful for both sides of the market [**?**].

In some contexts, interviews can help us determine the underlying preferences. Hence, one could simply conduct all pairwise interviews, determine agents' underlying strict preferences, and apply the algorithm of Gale and Shapely [**?**]. Interviews, however, can be expensive and so one may wish to minimize their number. In this work we investigate the extent to which we can learn about the employer-optimal stable matching of a given market, with partially ordered preferences, without performing any interviews.

We show that we can decide in polynomial time whether there exists a *pervasive employer-optimal matching*, i.e., a matching that is employer-optimal stable under all strict total orders that are consistent with the partial preference orderings. We provide a polynomial time algorithm to identify a pervasive employer-optimal matching, if it exists, or to report that none exists. Note that if a setting admits a pervasive employer-optimal matching, then no additional preference information need be elicited. A computationally tractable method for identifying such matchings is therefore useful in practice as a stopping condition for a procedure that repeatedly conducts interviews until the employer-optimal matching has been determined.

What if a setting does not admit a pervasive employer-optimal matching? In this case, it is still interesting to identify identical "components" among the matchings that are employer-optimal stable with respect to some underlying strict preferences. We show however that, even under quite severe restrictions on partial preference orderings, it is co-NP-complete to decide whether a given pair of agents constitute a *necessary match*; i.e., if they are matched in the employer-optimal stable matching of all underlying strict total preference ordering profiles. Similarly, we also show that it is co-NP-complete to decide whether it is *impossible* for a given pair of agents to be matched in the employer-optimal stable matchings of any underlying strict total preference ordering profile.

Lastly, we seek a better understanding of the relationship between matchings that are stable with respect to a given partial preference ordering and those that are employer-optimal with respect to some strict total order that refines partial preferences. We supply a polynomial-time algorithm for determining whether a given stable matching is also optimal for some such refinement.

Our partial information setting is relevant to the literature in which agents may declare indifference between candidates (see, e.g., **????**). This literature does not directly apply to our setting, as it assumes that an agent is truly indifferent between two or more candidates. However, it provides techniques that we build on to prove some of our results. Our setting is also relevant more broadly to preference reasoning and aggregation, where agents cast votes by providing their preferences over alternatives. In cases where it may be impractical for an agent to give a linear order over all alternatives, agents may be permitted to submit partial orders instead. **?** and **?** study such a setting; they ask whether an alternative c is a winner regardless of how partial orders are extended to linear orders, and, if not, whether c wins under some extension of the partial orders to linear orders.

The paper is organized as follows. We formally define our setting in Section 2. We present our polynomial-time algorithm for identifying whether a setting admits a pervasive employer-optimal matching in Section 3. Our hardness results regarding necessary and impossible matches are presented in Section 4. We present our polynomial-time algorithm for determining whether a given matching, stable with respect to a given partial preference ordering, is also optimal for some such refinement in Section 5. We discuss possible further extensions of our work in Section 6.

2. OUR MODEL

In a two-sided matching market, participants are partitioned into two disjoint sets, and each participant wishes to be matched to one or more participants from the other side of the market. Let $A = \{a_1, \ldots, a_n\}$ be a set of applicants and let $E = \{e_1, \ldots, e_m\}$ be a set of employers. We use the term *agents* when making statements that apply to both applicants and employers, and the term *candidates* to refer to agents on the side of the market opposite to that of an agent being considered. We assume that each employer can hire at most one applicant, and that each applicant can be hired by at most one employer, hence focusing on one-to-one matching markets. Following the model introduced in our own past work [**?**], we assume that agents start out only partially aware of their preferences. Formally, each agent is initially aware of a strict partial preference ordering over (a subset of) the candidates. We denote by p_{e_i} and p_{a_j} the strict partial preference ordering of e_i and a_j, respectively. We let $p_{E,A} = (p_{e_1}, \ldots, p_{e_m}, p_{a_1}, \ldots, p_{a_n})$ and call $p_{E,A}$ a *partial preference ordering profile*. For example, agents might start out by assigning candidates to equivalence classes, and having a strict preference ordering over these equivalence classes. This *equivalence class ordering* is a natural model for scenarios in which each agent knows that some candidates are her top-tier candidates, that others are her second-tier candidates, and so on. Figure 1 of Example 2.4 depicts such a setting, with each agent's partial preference ordering partitioning the candidates into strictly ranked equivalence classes.

Agents' true preferences are strict total orders: each applicant a has a strict preference ordering $\succ_a$ over $E \cup \{\varnothing\}$, where $\varnothing$ represents remaining unmatched, and each employer e has a strict preference ordering $\succ_e$ over $A \cup \{\varnothing\}$. If an agent i prefers $\varnothing$ to candidate j, we say j is *unacceptable* to i; all other candidates are *acceptable* to i.[1] We let $\succ_{E,A} = (\succ_{e_1}, \succ_{e_2}, \ldots, \succ_{e_m}, \succ_{a_1}, \succ_{a_2}, \ldots, \succ_{a_n})$ and call $\succ_{E,A}$ a *strict total preference ordering profile*, or just *strict preference profile*. The strict preference profiles $\succ_{E,A}$ are consistent with the partial preference ordering profile. That is to say, for each agent i and for every pair of candidates j and k: (i) if i prefers j to k according to the partial preference ordering profile then i strictly prefers j to k under $\succ$; and (ii) candidate j appears in i's partial preference ordering if and only if j is acceptable to i under $\succ$.

We denote by $I = (E, A, p_{E,A})$ an instance of a two-sided matching market. We say that an instance $I' = (E, A, p'_{E,A})$ *refines* another instance $I = (E, A, p_{E,A})$, which we write as $I' \lhd I$, if all strict preference profiles $\succ$ that are consistent with $p'_{E,A}$ are also consistent with $p_{E,A}$. We say that a strict preference profile $\succ$ refines an instance $I = (E, A, p_{E,A})$, and write $\succ \lhd I$, if $\succ$ is consistent with $p_{E,A}$. We say that an agent x *strictly prefers* (or *prefers*) a candidate y to another candidate z, w.r.t. a given instance $I = (E, A, p_{E,A})$ or w.r.t. a given partial preference ordering profile $p_{E,A}$, if and only if x prefers y to z under all strict preference profiles that are consistent with $p_{E,A}$. In this case, we also say that agent z is a successor to agent y in x's preference ordering. We say that x *weakly* prefers y to z, and denote this by $y \succeq_x z$, if and only if there exists a strict preference profile $\succ \lhd I$ under which x prefers y to z.

Throughout this paper we often refine an instance by *promoting* an agent y above another agent z in an agent x's preference ordering. That is, we refine the instance by ensuring that $y \succ_x z$. Promotions may be redundant: x may already (strictly) prefer y to z. For convenience and clarity, we adopt the convention that employers are male and applicants are female.

2.1. Optimal stable matchings

Our main interest is in matchings that are stable with respect to agents' underlying preferences and furthermore optimal for one side of the market. We now define these notions formally.

Definition 2.1 (*Matching*). A matching $\mu : A \cup E \mapsto A \cup E \cup \{\varnothing\}$ is an assignment of applicants to employers such that each applicant is assigned to at most one employer and vice versa. More formally, (i) $\mu(a_j) = e_i$ if and only if $\mu(e_i) = a_j$, (ii) for all $a_j \in A$ either there exists $e_i \in E$ such

[1] We assume that agents have strict preferences over unacceptable candidates only to simplify notation.

$\mathbf{e_1}$	$\mathbf{e_2}$
a_1	a_1
a_2	a_2

$\mathbf{a_1}$	$\mathbf{a_2}$
e_1	e_2
e_2	e_1

Fig. 1. A setting with 2 employers and 2 applicants. Applicants have full knowledge of their preferences; employers do not.

(a)

$\mathbf{e_1}$	$\mathbf{e_2}$
a_1	a_1
a_2	a_2

(b)

$\mathbf{e_1}$	$\mathbf{e_2}$
a_1	a_2
a_2	a_1

(c)

$\mathbf{e_1}$	$\mathbf{e_2}$
a_2	a_1
a_1	a_2

(d)

$\mathbf{e_1}$	$\mathbf{e_2}$
a_2	a_2
a_1	a_1

Fig. 2. The four possible underlying preference profiles for the employers in the setting of Table 1.

that $\mu(a_j) = e_i$ or $\mu(a_j) = \varnothing$ (the applicant is unmatched), and likewise (iii) for all $e_i \in E$ either there exists $a_j \in A$ such that $\mu(e_i) = a_j$ or $\mu(e_i) = \varnothing$.

An agent x *strictly prefers* (or *prefers*) a matching μ to another matching μ' if and only if $\mu(x) \succ_x \mu'(x)$. We say that x *weakly prefers* μ to μ' if $\mu(x) \succeq_x \mu'(x)$. A *maximum cardinality* (*maximum*) matching is a matching that matches the greatest number of agents.

Definition 2.2 (*Blocking pair*). A pair (a_j, e_i) is a *blocking pair* with respect to matching μ if a_j and e_i are not matched together in μ and they prefer each other to their assigned partners.

Definition 2.3 (*Stable matching*). A matching μ is *stable* with respect to a given instance I if it has no blocking pair and no agent is matched to an unacceptable partner.

Example 2.4. Consider the setting with 2 employers and 2 applicants depicted in Figure 1. The column corresponding to each agent i describes that agent's strict partial preference ordering, which is an equivalence class ordering; the most preferred equivalence class is at the top. In this example applicants have full knowledge of their preferences, while employers have no knowledge of their preferences. Figure 2 illustrates all four possible strict preference orderings for the employers. In every case, matching μ_1, $\mu_1(e_1) = a_1$ and $\mu_1(e_2) = a_2$, is stable. Matching μ_2, $\mu_2(e_1) = a_2$ and $\mu_2(e_2) = a_1$, is also stable under (c). Matching μ_2 is not stable in the other cases: (e_1, a_1) blocks μ_2 under (a) and (b), while (e_2, a_2) blocks μ_2 under (d).

Employer-optimal and applicant-optimal matchings are particularly interesting: these are the stable matchings most preferred by all employers (resp., applicants), as compared to all other stable matchings. When agents have strict preferences, as in our model, such matchings always exist and are unique [**?**]. However, an employer-optimal matching and/or an applicant-optimal matching may not exist if the true preferences of the agents are partially ordered. Employer-optimal matchings can be used to build mechanisms in which truthtelling is a dominant strategy for the employers [**?**]. Of course, this claim and all of our technical results can be made to apply instead to applicant-optimal matchings by swapping every use of the terms "employer" and "applicant".

Definition 2.5 (*Employer-optimal matching*). A matching is employer-optimal if it is stable and every employer weakly prefers it to every other stable matching.

Example 2.6. Continuing Example 2.4, μ_1 is the employer-optimal matching in cases (a), (b), and (d), and μ_2 is the employer-optimal matching in case (c).

In a similar vein, we can define the applicant-pessimal stable matching. A matching is employer-optimal if and only if it is applicant-pessimal [**??**].

Definition 2.7 (*Applicant-pessimal matching*). A matching μ is applicant-pessimal if it is stable and all applicants weakly prefer every other stable matching to μ.

3. PERVASIVE EMPLOYER-OPTIMAL MATCHINGS

We are interested in the following question. Given an instance $I = (E, A, p_{E,A})$, do all strict preference profiles that refine I have the same employer-optimal matching μ?

Definition 3.1 (*Pervasive employer-optimal (PEO) matching*). We say that matching μ is a *pervasive employer-optimal (PEO) matching* w.r.t. instance $I = (E, A, p_{E,A})$ if and only if μ is the employer-optimal matching w.r.t. all strict preference profiles that refine I.

We call I an *employer-optimal-unique* instance of the stable matching problem with partial information if and only if I admits a PEO matching.

We are going to show that we can find, in polynomial time, a PEO matching for I, or prove that none exists. In the proofs of our main claims, we build on results from the literature on stable matchings under preferences that include indifference. In this literature (see, e.g., **??**), the stable matching problem with ties, with incomplete lists (i.e. the agents are allowed to declare some candidates unacceptable), and with both ties and incomplete lists are referred to by SMT, SMI, and SMTI, respectively. For notational consistency, from now on we denote by SMP the stable matching problem with partial information, and by SMEC the stable matching problem with partial information structured as equivalence class orderings.

The remainder of this section is organized as follows. We first introduce the extensively studied notion of super-stability (see, e.g., **?**), that is closely related to our notion of pervasive employer-optimal matchings. We argue that if a given instance I of SMP admits a PEO matching μ, then μ must be the employer-optimal super-stable matching in I. Our polynomial-time algorithm IS-EMP-OPT-UNIQUE (outlined in Algorithm 1) is hence composed of two algorithms, SUPER-SMP (described in Section 3.2) and IS-PERVASIVE (described in Section 3.3). SUPER-SMP finds the employer-optimal super-stable matching Z of I, if I admits any super-stable matching. IS-PERVASIVE checks whether or not Z is a PEO matching for I.

Algorithm 1: IS-EMP-OPT-UNIQUE

```
Input: I = (E, A, p_{E,A})
Output: Matching μ that is PEO w.r.t. I, or 'false' if no such matching exists

Z = SUPER-SMP(I) ;
if Z = 'false' then
    return 'false';
else if IS-PERVASIVE(I, Z) then
    return Z ;
else
    return 'false';
```

3.1. Super-stable matchings

When indifference is allowed in the preference lists, it is not clear how one should define stability. Three versions of stability have been defined in the literature (see, e.g., **?**), including super-stability. Loosely speaking, super-stable matchings are stable no matter how indifferences in the preferences are resolved. Polynomial-time algorithms have been identified for finding such matchings if they exist or for reporting that none exists in various models of two-sided matching markets [**????**].

Definition 3.2 (*Super-stability*). A matching μ is *super-stable* if no agent is matched to an unacceptable partner and there is no unmatched pair (e_i, a_j) where each of them either strictly prefers the other to his/her partner in μ or is indifferent between them.

The notion of super-stable matchings can be extended to our partial information setting by interpreting incomparability as indifference. Therefore, we say that an agent x is indifferent between agents y and z, $y \neq z$, under a given partial preference ordering profile $p_{E,A}$ if x finds y and z incomparable. Observe that a matching is super-stable w.r.t. I if and only if it is stable w.r.t. all strict preference profiles that can be derived from $p_{E,A}$ by resolving indifferences (see, e.g. [**?**]). Hence, if a matching μ is to be the employer-optimal matching for every refinement of I, then μ must be a super-stable matching for I. It is worth noting that in the literature (see, e.g., **?**) indifference is not necessarily considered to be a transitive relation. In other words, it is permitted to say that an agent x is indifferent between agents y and z and also indifferent between agents z and q while strictly preferring y to q. Observe that a partial preference ordering in which indifference is transitive is an equivalence class ordering. Employer-optimal super-stable matchings can be defined analogously to employer-optimal stable matchings.

Definition 3.3 (*Employer-optimal super-stable matching*). A matching is employer-optimal super-stable if it is super-stable and it is weakly preferred by all employers to every other super-stable matching.

Super-stable matchings are not always guaranteed to exist. However, it is known that an employer-optimal super-stable matching exists whenever a super-stable matching exists [**?**].

PROPOSITION 3.4. *If an instance I admits a pervasive employer-optimal matching μ, then μ must be the employer-optimal super-stable matching in I.*

This claim, however, does not hold in the other direction. That is, the employer-optimal super-stable matching of I, if it exists, need not be a pervasive employer-optimal matching—in which case I does not admit a pervasive employer-optimal matching.

Example 3.5. Continuing Example 2.6, μ_1 is the one and only super-stable matching for the setting depicted in Figure 1, and hence is the employer-optimal super-stable matching for this setting. The employer-optimal matching for the strict preference profile in case (c) is μ_2. Therefore, μ_1 is not a PEO matching and hence the problem instance is not employer-optimal unique.

3.2. An Algorithm to find a super-stable matching in SMP

We now present SUPER-SMP, an algorithm that, given an instance $I = (E, A, p_{E,A})$ of SMP, identifies a super-stable matching for I if such a matching exists. It is an extension of the algorithms SUPER and SUPER2 by **?** and **?** for finding super-stable matchings in SMT and SMTI, which in turn are extensions of the deferred acceptance algorithm, proposed by Gale and Shapley [**?**], that finds the employer-optimal matching in SMI. SUPER-SMP finds the employer-optimal super-stable matching, if the set of super-stable matchings for I is not empty; otherwise, it reports that no super-stable matching exists.

SUPER-SMP is formally defined as Algorithm 2. Informally, the algorithm conducts a sequence of proposals by employers to applicants. After receiving one or more proposals, each applicant a (i) tentatively accepts a proposal she likes the best and becomes "engaged" to the corresponding employer, e, and (ii) rejects the rest of the proposals and declares all the employers she likes less than e to be unacceptable. An applicant a may receive two or more proposals that she likes the best and hence becomes multiply engaged. In this case, a rejects all the proposals and declares unacceptable (1) all the employers she likes less than one or more of her best proposals, as well as (2) those she cannot compare with at least one of her best proposals. An employer e proposes to an applicant a if (i) e and a are not already engaged, (ii) they are acceptable to each other, and (iii) there is no applicant a' that finds e acceptable whom e strictly prefers to a.

The algorithm halts when no more proposals can be made. Let G be the engagement relation at the time when the algorithm halts. Let μ be a maximum cardinality matching in G. If there is some applicant a who has received a proposal (during the execution of the algorithm) but is not matched in μ, then no super-stable matching exists. Otherwise, μ is the employer-optimal super-stable matching.

When we say *delete* the pair (e, a), we mean that e should be deleted from the preference ordering of a and that a should be deleted from the preference ordering of e. For any agent x, we refer to x's preference ordering at the termination of SUPER-SMP as x's *reduced preference ordering*.

At any stage of the algorithm, we say that an applicant a is at the *head* of an employer e's preference ordering if there is no other applicant in e's remaining preference ordering whom e strictly prefers to a. Note that more than one applicant can be at the head of e's preference ordering.

Algorithm 2: SUPER-SMP

```
Input: I = (E, A, p_{E,A})
Output: Matching μ that is the employer-optimal super-stable matching of I, or 'false'
Initialize
    proposed(a) ← false, ∀a ∈ A ;
repeat
  Main loop
    while some employer e has an applicant at the head of his preference ordering to whom he is
    not engaged do
        foreach applicant a at the head of e's preference ordering do
            e proposes, and becomes engaged, to a ;
            proposed(a) ← true ;
            foreach strict successor e' of e on a's preference ordering do
                if e' is engaged to a then
                    break the engagement ;
                delete the pair (e', a);
    foreach applicant a who is multiply engaged do
        foreach employer e that a likes less than, or cannot compare with, at least one of her
        fiancées do
            delete the pair (e, a);
        break all engagements involving a ;
until each employer is either engaged to all applicants at the head of his preference ordering, or
has an empty preference ordering;
let μ be a maximum cardinality matching in the engagement relation ;
Last check
if some applicant a is unmatched in μ and proposed(a) then
    return 'false';
else
    return μ ;
```

To prove the correctness of SUPER-SMP, we follow an approach similar to that taken in **?** and **?**. The following lemmas are useful in proving the main claim of this section, that SUPER-SMP is sound and complete, and are also interesting in their own right. The proofs, except for those of Lemma 3.8 and Corollary 3.10, are quite similar to the proofs of similar claims in **?** and **?** and hence removed.

Informally, our first claim states that for a given instance of SMP, the set of agents may be partitioned into two subsets, those matched in all super-stable matchings, and those matched in none.

LEMMA 3.6. *For a given instance of SMP, let μ and μ' be two super-stable matchings. Then for any agent x in the instance, x is matched in μ if and only if x is matched in μ'.*

LEMMA 3.7. *If the pair (e, a) is deleted during an execution of SUPER-SMP, then that pair cannot block any matching output by SUPER-SMP.*

LEMMA 3.8. *A matching output by SUPER-SMP is super-stable.*

PROOF. Suppose, for a contradiction, that some execution of SUPER-SMP outputs a matching μ that is blocked by some pair (e, a). Thus e and a are acceptable to each other. By Lemma 3.7, the pair (e, a) has not been deleted, hence each is on the reduced preference ordering of the other.

Let G be the engagement relation at the termination of the algorithm. Let E_G and A_G denote the sets of employers and applicants engaged under G, respectively. Note that each applicant has degree at most one in G; therefore each employer $e \in E_G$ is matched in μ (or μ is not maximum). Also, each applicant $a \in A_G$ is matched in μ, as otherwise the algorithm reports that no super-stable matching exists, a contradiction. Since all agents engaged under G are matched, it must be the case that $|E_G| = |A_G|$. Furthermore, since each applicant has degree at most one in G, it must also be the case that each employer has degree at most one in G. Thus G is a one-to-one relation.

Since a is on the reduced preference ordering of e, it follows that e's reduced preference ordering is nonempty and hence e must be engaged in G; let us call his fiancée a'. By the argument just given, e must be matched to a' in μ. Since (e, a) blocks μ, it follows that $a \neq a'$. If e strictly prefers a to a', then the pair (e, a) has been deleted, since a' is at the head of the reduced preference ordering of e, a contradiction. Thus e cannot compare a and a'.

If a is at the head of the reduced preference ordering of e, then e must have proposed to a and hence must be engaged to her in G, as otherwise (e, a) must have been deleted, a contradiction. Since a is engaged to e, and only to e, in G and has a partner in μ, thus she must be matched in μ to e. Therefore (e, a) is in μ and cannot block μ, a contradiction.

Thus a is not at the head of the reduced preference ordering of e and therefore there is an applicant whom e strictly prefers to a. Since I is a strict partial order and there is no cycle in the preference relation, there must be an applicant at the head of e's reduced preference ordering, say a^*, whom e strictly prefers to a. We have already established that e cannot compare a and a', therefore $a^* \neq a'$. Since a^* is at the head of the reduced preference ordering of e, e must have proposed to a^* and consequently must be engaged to her in G, as otherwise (e, a^*) must have been deleted and a^* cannot be in e's reduced preference ordering, a contradiction. Therefore e is engaged to at least two applicants, a and a^*, contradicting the previously established fact that G is a one-to-one relation. □

We say that a pair (e, a) is super-stable if e and a are matched in a super-stable matching.

LEMMA 3.9. *No super-stable pair is ever deleted during an execution of SUPER-SMP.*

COROLLARY 3.10. *The matching that SUPER-SMP outputs is the employer-optimal super-stable matching.*

PROOF. By Lemma 3.8 SUPER-SMP outputs a super-stable matching. Let μ be the super-stable matching output by SUPER-SMP. Assume for contradiction that μ is not the employer-optimal super-stable matching Z. Thus there must be an employer, say e, who is matched in Z and strictly prefers $Z(e)$ to his match in μ. Thus, $(e, Z(e))$ must have been deleted during the execution of SUPER-SMP for e to propose to $\mu(e)$. (Note that, by Lemma 3.6, since e is matched in Z he must also be matched in μ.) However, by Lemma 3.9, no super-stable pair is deleted during the execution of SUPER-SMP, a contradiction. □

Our next claim shows that the last check at the end of SUPER-SMP correctly identifies the cases where no super-stable matching exists. Theorem 3.12 then shows that SUPER-SMP is sound and complete.

LEMMA 3.11. *If some applicant receives a proposal and is unmatched in the maximum matching* μ *, then no super-stable matching exists for the given instance.*

THEOREM 3.12. *For a given instance* I *of SMP, SUPER-SMP determines, in polynomial time, whether or not a super-stable matching exists. If such a matching does exist, all possible executions of the algorithm find the employer-optimal super-stable matching of* I.

3.3. An algorithm to check for a pervasive employer-optimal matching

In this section we present IS-PERVASIVE, an algorithm that, given an instance I of SMP that admits super-stable matchings, and Z, the employer-optimal super-stable matching of I, decides in polynomial time whether Z is the employer-optimal matching for every refinement of I.

Recall that, for an instance of SMI, a matching is employer-optimal if and only if it is applicant-pessimal. **?** showed that a matching μ is applicant-pessimal if there is no rotation exposed in μ (Lemma 2.5.3 in [**?**]).[2] We are going to use this notion in the main proof of this section, so we provide a brief definition of an *exposed rotation* here.

Definition 3.13 (*Exposed rotation*). Let μ be a stable matching for an instance I of SMI. For each applicant a, we define the *next-best* employer for a relative to μ. Let $\eta_\mu(a)$ denote the first employer e on a's preference ordering after $\mu(a)$ such that e prefers a to $\mu(e)$. If a is not matched to $\mu(a)$ in μ_e, the employer-optimal matching of I, then $\eta_\mu(a)$ must exist, since $\mu_e(a)$ qualifies. Otherwise, $\eta_\mu(a)$ may be undefined. A *rotation exposed* in μ is a cyclic sequence of pairs $(a_{i_0}, e_{i_0}), ..., (a_{i_{r-1}}, e_{i_{r-1}})$ such that $\mu(a_{i_j}) = e_{i_j}$ and $\eta_\mu(a_{i_j}) = e_{i_{j+1}}$, $\forall\, j$, $j+1$ taken modulo r.

IS-PERVASIVE (outlined in Algorithm 3) first constructs a directed graph $G(I)$ that facilitates the search for potential rotations in Z. A vertex in G is created for each applicant in the instance, and an edge $(a, Z(e))$ is created for each employer $e \in \mathcal{M}(a)$, where e is in $\mathcal{M}(a)$ if and only if (1) $Z(a) \succ_a e$, and (2) $a \succeq_e Z(e)$, and (3) there is no employer e' such that $Z(a) \succ_a e'$, $e' \succ_a e$ and $a \succ_{e'} Z(e')$. In other words, $e \in \mathcal{M}(a)$ if and only if (1) e is a successor to $Z(a)$ in a's preference ordering, and (2) e either strictly prefers a to $Z(e)$ or is indifferent between them, and (3) there is no other employer e' whom a strictly ranks between $Z(a)$ and e, such that e' strictly prefers a to $Z(e')$.

Algorithm 3: IS-PERVASIVE

Input: $I = (E, A, p_{E,A})$ and matching Z such that Z is the employer-optimal super-stable matching of I
Output: 'true' if Z is pervasive employer-optimal w.r.t. I, and 'false' otherwise

Create $G(I)$;
if $G(I)$ *is acyclic* **then**
  return *'true'*;
else
  return *'false'*;

THEOREM 3.14. *For a given instance I of SMP that admits a super-stable matching, the employer-optimal super-stable matching Z is pervasive if and only if the graph $G(I)$ is acyclic.*

PROOF. We first show that if $G(I)$ contains a cycle then there is an SMI instance I' that is a refinement of I for which Z is not the employer-optimal matching. Suppose that $(a_{i_0}, a_{i_1}, \ldots, a_{i_{r-1}})$ is a cycle in $G(I)$, and let $e_{i_j} = Z(a_{i_j})$ for $0 \le j \le r-1$. We break indifference in I to construct I' in the following way. (Here and elsewhere, subscripts are taken modulo r.)

— In a_{i_j}'s preference ordering, promote $e_{i_{j+1}}$ ahead of any employer with whom he is incomparable;
— in $e_{i_{j+1}}$'s preference ordering, promote a_{i_j} ahead of any applicant with whom she is incomparable;
— in the preference ordering of any other employer e, promote $Z(e)$ ahead of any applicant with whom she is incomparable;
— resolve all other indifferences arbitrarily.

[2]Lemma 2.5.3 in [**?**] is stated for SM, but the result extends easily to SMI.

Define the matching μ so that $\mu(a_{i_j}) = e_{i_{j+1}}$ for $0 \leq j \leq r-1$ and $\mu(a) = Z(a)$ for all other applicants a. It follows from the precedence given to a_{i_j} in indifference-breaking $e_{i_{j+1}}$'s preference ordering that, in I', no employer is worse off in μ as compared to Z. We claim that μ is stable in I', and thus Z cannot be the employer-optimal matching for I'.

Suppose that (e, a) blocks μ in I'. If $\mu(a) = Z(a)$, then (e, a) blocks Z in I', contradicting the super-stability of Z. So $a = a_{i_j}$ for some j. But $\mu(a_{i_j}) = e_{i_{j+1}}$, so a_{i_j} prefers e to $e_{i_{j+1}}$ in I', and e prefers a_{i_j} to $\mu(e)$ in both I and I', and hence also to $Z(e)$ in I'. If, in I', a_{i_j} prefers e to e_{i_j}, then (a_{i_j}, e) blocks Z, contradicting the super-stability of Z. Otherwise, because of the precedence given to $e_{i_{j+1}}$ in refining a_{i_j}'s preference ordering, a_{i_j} must prefer e to $e_{i_{j+1}}$ in I, but this contradicts condition 3 used in defining $\mathcal{M}(a)$ for the construction of $G(I)$.

Now suppose that there is a refinement I' of I for which Z is not the employer-optimal matching, and hence not the applicant-pessimal matching. It follows that, w.r.t. Z in I' there is an exposed rotation $(a_{i_0}, e_{i_0}), \ldots, (a_{i_{r-1}}, e_{i_{r-1}})$. We claim that $(a_{i_0}, \ldots, a_{i_{r-1}})$ forms a cycle in $G(I)$.

According to the definition of a rotation, $e_{i_{j+1}}$ is the first successor e of e_{i_j} on a_{i_j}'s preference ordering in I' such that e prefers a_{i_j} to $Z(e)$ in I'— and thus in I either e strictly prefers a_{i_j} to $Z(e)$ or is indifferent between them. It follows that $e_{i_{j+1}} \in \mathcal{M}(a_{i_j})$. If this were not the case, then, in I, a_{i_j} prefers $e_{i_{j+1}}$ to $e_{i_j} = Z(a_{i_j})$ or is indifferent between them. But, in I', $e_{i_{j+1}}$ prefers a_{i_j} to $a_{i_{j+1}} = Z(e_{i_{j+1}})$ and so, in I, $e_{i_{j+1}}$ prefers a_{i_j} to $a_{i_{j+1}} = Z(e_{i_{j+1}})$ or is indifferent between them. It follows that the pair $(a_{i_j}, e_{i_{j+1}})$ blocks Z as a super-stable matching, a contradiction. Therefore, $(a_{i_j}, a_{i_{j+1}})$ is an edge of $G(I)$. Hence the rotation yields a cycle in $G(I)$. □

THEOREM 3.15. *Given an instance I of SMP, IS-EMP-OPT-UNIQUE determines, in polynomial time, whether I is an employer-optimal-unique instance and, if it is, returns the pervasive employer-optimal matching.*

PROOF. IS-EMP-OPT-UNIQUE (Algorithm 1) first calls SUPER-SMP. If SUPER-SMP returns 'false'—i.e. no super-stable matching exists, then IS-EMP-OPT-UNIQUE returns 'false'—i.e. no PEO matching exists. Otherwise, SUPER-SMP returns the employer-optimal super-stable matching Z of I. IS-PERVASIVE is called next which constructs $G(I)$, as described above, and checks whether $G(I)$ is acyclic. If so, IS-EMP-OPT-UNIQUE returns Z; otherwise, it returns 'false'.

We first verify the correctness of IS-EMP-OPT-UNIQUE. Note that, by Theorem 3.12, SUPER-SMP returns the employer-optimal super-stable matching of I, Z, if I admits super-stable matchings. Following Proposition 3.4, I cannot admit a pervasive employer-optimal matching if it does not admit one or more super-stable matchings. Furthermore, by Theorem 3.14, Z is pervasive if and only if $G(I)$ is acyclic. It remains to show that the above steps can be performed in polynomial time. By Theorem 3.12, SUPER-SMP halts in polynomial time. It is straightforward to verify that the construction of the directed graph $G(I)$ takes $O(n \cdot m)$ time, as does testing whether $G(I)$ contains a cycle. Hence the overall algorithm terminates in polynomial time. □

4. NECESSARY AND IMPOSSIBLE MATCHES

In the previous section, we showed how to efficiently decide whether a given instance I is employer-optimal unique. What if I does not admit a pervasive employer-optimal matching? Can we still determine whether there exist identical "components" among the matchings that are employer-optimal w.r.t. some refinement of I? Here we tackle this question by studying the following two closely related problems. First, we investigate the problem of identifying pairs that are *never* matched in the employer-optimal matching for *any* of the strict preference profiles that refine I. Next, we study the problem of identifying pairs that are *always* matched in the employer-optimal matching for *all* of the strict preference profiles that refine I.

Definition 4.1 (*Impossible matches*). Given I, a pair (e, a) is an *impossible* match if for all $\succ$ that refine I, e and a are not matched in the employer-optimal matching of $\succ$.

Definition 4.2 (*Necessary matches*). Given I, a pair (e, a) is a *necessary* match if for all $\succ$ that refine I, e and a are matched in the employer-optimal matching of $\succ$.

4.1. Hardness of identifying impossible matches

The proof of our next claim, on the hardness of identifying impossible matches, uses the same construction as in a hardness proof (Theorem 5.1) of **?** for a related problem. We first define some useful terms from **?**.

A *master* list of employers consists of a single list containing all of the employers, which may or may not contain ties. We will consider settings in which each applicant's preference ordering contains her acceptable partners ranked precisely according to such a master list—as if each applicant takes the master list of employers and removes those she finds unacceptable. A master list of applicants and resulting employers' preferences are defined analogously. Extensions SMTI-2ML and SMTI-1ML denote problem variants involving master lists on both sides and on one side of the market respectively. For example, SMTI-2ML represents the Stable Matching problem with Ties and Incomplete Lists, with a Master List on both sides.

? show that COMPLETE SMTI-2ML, the problem of deciding whether a given instance of SMTI-2ML admits a complete stable matching, is NP-complete even if ties occur only in the master list of applicants. We prove that identifying impossible matches is hard by reducing from this problem.

THEOREM 4.3. *For a given instance of SMEC, and a given (employer, applicant) pair* (e, a)*, it is co-NP-complete to decide whether* (e, a) *is an impossible match, even if all applicants have the same equivalence classes and all agents are acceptable to the agents on the other side of the market.*

PROOF. The problem is in co-NP because the following nondeterministic polynomial-time algorithm determines that (e, a) is not an impossible match. Simply guess a strict preference profile $\succ$, verify that $\succ$ is consistent with the given SMEC instance and, by using the Gale-Shapley algorithm, that e and a are matched in the employer-optimal matching of $\succ$.

To prove co-NP-hardness, we show that the complement of the problem is NP-hard by reducing from the variant of COMPLETE SMTI-2ML in which ties occur only in the master list of applicants, which is NP-complete by Theorem 3.1 in **?**. Let J be an instance of this problem, where L_e and L_a are the master lists of employers and applicants respectively. Let $E = \{e_1, \ldots, e_n\}$ be the set of employers and $A = \{a_1, \ldots, a_n\}$ be the set of applicants in J. Let Q_{e_i} be the preference ordering of each employer e_i under J. (Note that Q_{e_i} may include ties.) We construct an instance I of our problem. Let the set of employers in I be $E \cup \{e_0\}$ and the set of applicants be $A \cup \{a_0\}$. The partial preference ordering for each agent in I is as follows.

$$
\begin{array}{ll}
e_0 : -- \, a_0 & \qquad a_0 : L_e \, e_0 \\
e_i : Q_{e_i} \, a_0 \, -- & \qquad a_i : L_e \, e_0
\end{array}
$$

In a given agent's partial preference ordering, we use the symbol $--$ to denote all remaining agents on the opposite side of the market in arbitrary strict order. (Note that an agent may not find all candidates acceptable under J.) Clearly all applicants have the same equivalence classes with e_0 being the only member of the lowest ranked equivalence class. We show that J admits a complete stable matching if and only if (e_0, a_0) is not an impossible match in I.

Suppose that μ is a complete stable matching for J. Let $\mu' = \mu \cup \{(e_0, a_0)\}$. We prove that μ' is stable w.r.t. some strict preference profile $\succ$ that refines I. Then we show that (e_0, a_0) belongs to the employer-optimal matching of $\succ$, which proves that (e_0, a_0) is not an impossible match. Let $\succ$ be a strict preference profile where each agent ranks his partner under μ' at the top of his/her corresponding class. To see that μ' is stable w.r.t. $\succ$, note that neither (e_0, a_i) nor (e_i, a_0) can block μ'. It remains to show that (e_i, a_j), $1 \leq i, j \leq n$, cannot block μ'. To see this, note that for (e_i, a_j) to be a blocking pair (1) a_j must be in Q_{e_i}, since $\mu'(e_i)$ is, (2) e_i must rank a_j in a higher equivalence class than $\mu'(e_i)$, and (3) a_j must rank a_i in a higher equivalence class than $\mu'(a_j)$. However this means that under J both e_i and a_j strictly prefer each other to their partners under μ, thus they block μ, a contradiction. Let μ'_e denote the employer-optimal matching of $\succ$. By definition, every employer must like his partner under μ'_e at least as well as his partner under μ'. Assume for a contradiction that $(e_0, a_0) \notin \mu'_e$. Since μ'_e is a complete matching (which it must be as the preference orderings are complete), then $(e_i, a_0) \in \mu'_e$ for some e_i, $1 \leq i \leq n$. However e_i strictly prefers $\mu'(e_i)$ to a_0—his match under the employer-optimal matching, a contradiction.

Conversely, assume that there exists a strict preference profile $\succ$ that refines I such that (e_0, a_0) belongs to $\succ$'s employer-optimal matching, μ'. Clearly μ' is a complete matching for I and so is μ for J. It remains to show that μ is stable under J. Note that μ matches every employer e_i to an acceptable partner under J (and thus every applicant a_j to an acceptable partner as well). For if not, then e_i strictly prefers a_0 to $\mu'(e_i)$. Since a_0 strictly prefers e_i to e_0—who is her partner under μ'—then (e_i, a_0) blocks μ', a contradiction. Clearly μ is stable in J, as otherwise any pair that blocks μ in J would also block μ' in I. □

4.2. Hardness of identifying necessary matches

In this section we show that the problem of deciding whether a given pair is a necessary match is co-NP-complete. The next theorem will come in handy in proving the aforementioned hardness result. Let ANY-STABLE SMTI-1ML denote the problem of deciding, given an instance of SMTI-1ML and an applicant a, whether there is a stable matching in which a is matched.

THEOREM 4.4. *The ANY-STABLE SMTI-1ML problem is NP-complete.*

PROOF. The structure of the proof is similar to that of Theorem 5.1 in **?**. To see that the problem is in NP, we describe a nondeterministic polynomial-time algorithm to determine that a given applicant a is matched in a stable matching of a given instance of SMTI-1ML. The algorithm simply guesses a matching μ and verifies its stability by examining all potential blocking pairs and furthermore checking that each agent is matched to an acceptable partner. To show NP-hardness, we transform from COMPLETE SMTI-2ML which is shown to be NP-complete by Theorem 3.1 in **?**. Let J be an instance of this problem where L_e denotes the master list of employers. Let $E = \{e_1, \ldots, e_n\}$ be the set of employers and $A = \{a_1, \ldots, a_n\}$ be the set of applicants in J. Let Q_{e_i} and Q_{a_j} denote the preference ordering of e_i and a_j under J, respectively. We construct an instance J' of our problem as follows. Let the set of employers in J' be $E \cup \{e_0\}$, the set of applicants be $A \cup \{a_0\}$, and the preference ordering for each agent be as follows.

$$\begin{array}{ll} e_0 : - - a_0 & \qquad\qquad a_0 : e_0 \\ e_i : Q_{e_i} & \qquad\qquad a_i : Q_{a_i}\ e_0 \end{array}$$

We obtain a master list L'_e of employers in J' by appending e_0 to L_e. We show that J has a complete stable matching if and only if there is a stable matching for J' in which a_0 is matched.

Suppose that μ is a complete stable matching for J. Let $\mu' = \mu \cup \{(e_0, a_0)\}$. We claim that μ' is a stable matching in J'. To see this, first note that (e_i, a_0) cannot block μ', as e_0 is the only employer acceptable to a_0. Furthermore, (e_i, a_j), $1 \leq i, j \leq n$, cannot block μ' as otherwise the pair would also block μ under J. It remains to show that (e_0, a_j), $1 \leq j \leq n$ cannot block μ'. To see this, note that all applicants in A are matched in μ, and hence μ', and they all prefer their partners to e_0.

Conversely, assume that J' has a stable matching μ' such that $(e_0, a_0) \in \mu'$. We prove that $\mu = \mu' \setminus \{(e_0, a_0)\}$ is a complete stable matching for J. Clearly μ is stable under J, otherwise any pair that blocks μ would also block μ'. It remains to show that μ is a complete matching. Assume for contradiction that it is not. Then there exists an applicant a_j, $1 \leq j \leq n$, who is unmatched under both μ and μ'. However, e_0 is acceptable to all applicants, and so to a_j, and prefers all applicants in A to a_0. Therefore, (e_0, a_j) blocks μ', contradicting the assumption that μ' is stable. □

We can now show that identifying necessary matches is hard. To prove this claim, we reduce from ANY-STABLE SMTI-1ML, which we just proved is NP-complete.

THEOREM 4.5. *For a given instance of SMEC, and a given (employer, applicant) pair (e, a), it is co-NP-complete to decide whether (e, a) is a necessary match, even if all applicants have the same equivalence classes and all agents are acceptable to the agents on the other side of the market.*

PROOF. To see that the problem is in co-NP, we describe a nondeterministic polynomial-time algorithm for determining that (e, a) is not a necessary match. The algorithm simply guesses a strict preference profile $\succ$, verifies that $\succ$ is consistent with the given SMEC instance and, by using the Gale-Shapley algorithm, that e and a are not matched in the employer-optimal matching of $\succ$. To prove co-NP-hardness, we show that the complement of the problem is NP-hard by reducing from ANY-STABLE SMTI-1ML, which is NP-complete by Theorem 4.4. Let J be an instance of this problem, with L_e denoting the master list of employers. Let $E = \{e_1, \ldots, e_n\}$ be the set of employers and $A = \{a_1, \ldots, a_n\}$ be the set of applicants in J. Let Q_{e_i} denote the preference ordering of e_i and Q_{a_j} denote the preference ordering of a_j under J. Without loss of generality, we can assume that Q_{e_i} does not contain an applicant who finds e_i unacceptable, and similarly that Q_{a_j} does not contain an employer who finds a_j unacceptable. Let $a^* \in A$ be any given applicant in this setting. We construct an instance I of our problem as follows. Let the set of employers in I be $E \cup \{e^*\}$ and the set of applicants be A. The preference ordering for each agent in J' is as follows.

$$
\begin{aligned}
&e^* : a^* \\
&e_i : Q_{e_i} \;\; \forall e_i \in E, e_i \neq e^* \\
&a_i : L_e e^* \;\; \forall a_i \in A
\end{aligned}
$$

We obtain the identical equivalence classes for the applicants by appending e^* to the end of L_e.

Note that Q_{e_i} does not contain applicants who find e_i unacceptable (in addition to not containing applicants whom e_i finds unacceptable). Furthermore, e^* finds all applicants except a^* unacceptable. Hence the preference ordering of each applicant $a_i \in A$, $a_i \neq a^*$, under I essentially reduces to Q_{a_i}, and the preference ordering of a^* reduces to $Q_{a^*} e^*$.

It is then straightforward to verify that e^* and a^* are matched in all stable matchings for I, and hence in the employer-optimal matching for all refinements of I, if and only if a^* is unmatched in all stable matchings for J. Hence (e^*, a^*) is not a necessary match if and only if the answer to ANY-STABLE SMTI-1ML given J and a^* is yes. So the co-NP-hardness of our problem follows from the NP-completeness of ANY-STABLE SMTI-1ML. □

5. WHEN CAN A STABLE MATCHING BE EMPLOYER-OPTIMAL?

Let I be an instance of SMP and let μ be a matching that is stable w.r.t. I. We are interested in determining whether there exists a strict preference profile $\succ \lhd I$ such that μ is employer-optimal w.r.t. $\succ$. We present a polynomial-time algorithm, IS-EMP-OPT, that finds such a strict preference profile if it exists and reports that none exists otherwise. Before proceeding with the description of our algorithm, we present the following lemma, which will come in handy.

LEMMA 5.1. *Assume that μ is employer-optimal w.r.t. some strict preference profile that refines I. Then there exists a strict preference profile $\succ \lhd I$ such that μ is employer-optimal w.r.t. $\succ$ and each applicant a who is matched in μ ranks $\mu(a)$ ahead of all employers who are incomparable with $\mu(a)$ in a's partial preference ordering in I.*

PROOF. Assume that μ is employer-optimal w.r.t. $\succ' \lhd I$. Let $\succ \lhd I$ be a strict preference profile similar to $\succ'$ except that for each applicant a, $\mu(a)$ is moved up in a's total order to be ranked ahead of all employers who are incomparable with him in a's partial preference ordering in I. We show that μ is employer-optimal w.r.t. $\succ$.

Assume for contradiction that this is not the case. It is easy to see that μ must be stable under $\succ$ or it cannot be stable under $\succ'$. Hence, since we assumed that μ is not employer-optimal w.r.t. $\succ$, there must be a rotation exposed in μ (see Section 3.3). Let (a_{i_0}, e_{i_0}), ..., $(a_{i_{r-1}}, e_{i_{r-1}})$ be such a rotation. Following the definition of an exposed rotation, e_{i_k} prefers $a_{i_{k-1}}$ to $\mu(e_{i_k})$ under $\succ$ (and hence under $\succ'$), and a_{i_k} prefers $\mu(a_{i_k})$ to $e_{i_{k+1}}$, $\forall k$, $k+1$ taken modulo r. However, as there is no rotation exposed in μ under $\succ'$, $a_{i_{k-1}}$ must prefer e_{i_k} to $\mu(a_{i_{k-1}})$ under $\succ'$, for some $k-1$, k taken modulo r. Therefore, under $\succ'$, e_{i_k} prefers $a_{i_{k-1}}$ to $\mu(e_{i_k})$ and $a_{i_{k-1}}$ prefers e_{i_k} to $\mu(a_{i_{k-1}})$. Thus $(e_{i_k}, a_{i_{k-1}})$ blocks μ under $\succ'$, and hence μ cannot be employer-optimal w.r.t. $\succ'$, a contradiction. □

Algorithm 4 formally defines IS-EMP-OPT. As stated earlier in Section 3.3, μ is employer-optimal for a strict preference profile $\succ$ if there is no rotation exposed in μ. IS-EMP-OPT gradually refines I in order to rule out the existence of possible rotations, while ensuring that μ is stable for the refined instance. It terminates either by reaching an SMP instance I' such that for all $\succ \lhd I'$ there is no rotation exposed in μ, or by identifying that a rotation is exposed in μ for all $\succ \lhd I$. Agents are labeled throughout the algorithm. As we will prove later (Lemma 5.7), an agent x is labeled *good* whenever it is established that, no matter how indifferences are resolved in the current SMP instance, x cannot belong to any exposed rotation.

IS-EMP-OPT proceeds in seven steps. An intuitive description of the steps follows.

Step 1: *Label agents who are unmatched in μ good and label those who are matched bad.* Note that unmatched agents cannot be in an exposed rotation.

Step 2: *Perform refinements that are essential if μ is to be stable.*

Step 3: *For each applicant a matched under μ, promote $\mu(a)$ ahead of all employers e who are incomparable with $\mu(a)$ in a's preference ordering.* Lemma 5.1 justifies this step, which in turn simplifies the overall proof.

The goal of the next three steps is to refine the current SMP instance so as to avoid exposed rotations, if possible.

Step 4: *For each applicant a matched under μ, let $S(a)$ be the set of employers e who prefer a to $\mu(e)$. If $S(a)$ is empty, (i) for each employer e who is matched under μ and such that a prefers $\mu(a)$ to e, promote $\mu(e)$ ahead of a in the preference ordering of e, and (ii) label a and $\mu(a)$ good.* If no employer strictly prefers a to his partner under μ, then it is possible to refine the current SMP instance such that under all remaining strict preference profiles all employers e prefer $\mu(e)$ to a. As a result, $\eta_\mu(a)$ is undefined under all remaining strict preference profiles and hence a, and therefore $\mu(a)$, cannot contribute to an exposed rotation.

Step 5: *For each applicant a who is labeled bad, identify those employers e such that there exists a strict preference profile $\succ$ that refines the current SMP instance and under which e is $\eta_\mu(a)$. The set of such employers is denoted by $T(a)$.* No employer e whom a ranks inferior to one or more employers in $S(a)$ can be $\eta_\mu(a)$ under any strict preference profile that refines the current SMP.

Step 6: *While there is at least one bad applicant, attempt to identify a bad applicant a who has a good employer in $T(a)$. If there does not exist at least one such bad applicant, an exposed rotation is inevitable and the algorithm returns "false'. Otherwise, let a be a bad applicant with a good employer e in $T(a)$. Refine the preference orderings of a, e and the rest of the employers in $T(a)$ such that e is $\eta_\mu(a)$ under all strict preference profiles that refine the newly reached SMP. Label a and $\mu(a)$ good.*

Step 7: *Resolve the remaining indifferences arbitrarily and return the resulting strict preference profile $\succ$.*

We need to prove that IS-EMP-OPT is sound and complete. Let I_i denote the SMP instance at the end of step i. Recall, from Definition 3.13, that for a strict preference profile $\succ$, $\eta_\mu(a)$ denotes the first employer e on applicant a's preference ordering who prefers a to his match under μ. We first claim that if our algorithm returns a strict preference profile $\succ$, then μ is guaranteed to be stable w.r.t. $\succ$. (The proof is straightforward and hence removed to save space.)

LEMMA 5.2. *Let μ be a matching stable w.r.t. I. Then μ is stable w.r.t. all strict preference profiles $\succ \lhd I_3$. Furthermore, $\forall \succ \lhd I$ such that μ is stable w.r.t. $\succ$, it must be the case that $\succ \lhd I_2$.*

We next show that our definitions of $S(a)$ and $T(a)$ (in Steps 4 and 5 of Algorithm 4) accord with the intuition provided earlier.

LEMMA 5.3. *Let $\succ$ be a strict preference profile that refines I_3. For all applicants a and for all $e \in S(a)$, a strictly prefers $\mu(a)$ to e.*

Algorithm 4: IS-EMP-OPT

Input: $I = (E, A, p_{E,A})$ and matching μ stable w.r.t. I
Output: Strict total preference ordering profile $\succ \lhd I$ such that μ is employer-optimal w.r.t. $\succ$, or 'false'

Step 1 **Label** unmatched agents good and matched agents bad;

Step 2 `/* make promotions in preferences to ensure that` μ `is stable */`

foreach *agent x who is matched in μ* **do**
    foreach *candidate y ranked above $\mu(x)$ by x* **do**
        promote $\mu(y)$ ahead of x in y's preferences;

foreach *agent x who is unmatched in μ* **do**
    foreach *candidate y ranked by x* **do**
        promote $\mu(y)$ ahead of x in y's preferences;

Step 3 **foreach** *applicant a who is matched in μ* **do**
    promote $\mu(a)$, in a's preferences, ahead of all e such that a is indifferent between e and $\mu(a)$;

Step 4 **foreach** *applicant a who is matched in μ* **do**
    $S(a)$ = set of employers e who strictly prefer a to $\mu(e)$;
    if $S(a)$ *is empty* **then**
        Label a and $\mu(a)$ good;
        foreach *matched employer e such that a prefers $\mu(a)$ to e* **do**
            promote $\mu(e)$ ahead of a in e's preferences;

Step 5 `/* identify employers` e `who are` $\eta_\mu(a)$ `under some` $\succ$ `*/`

foreach *applicant a who is labeled bad* **do**
    $T(a)$ = set of employers e such that e weakly prefers a to $\mu(e)$ and a strictly prefers $\mu(a)$ to e and there is no employer $e' \in S(a)$ such that a strictly prefers e' to e.

Step 6 **while** *there is at least one bad applicant* **do**
    if *there is a bad applicant a who has a good employer e in $T(a)$* **then**
        promote e ahead of all employers e' such that a is indifferent between e' and e;
        if *e is indifferent between a and $\mu(e)$* **then**
            promote a ahead of $\mu(e)$ in e's preferences;
        foreach *employer $e' \in T(a)$ such that a prefers e' to e* **do**
            promote $\mu(e')$ ahead of a in the preferences of e';
        Label a and $\mu(a)$ good;
    else `/* cannot avoid a rotation ==> a stable matching that dominates` μ `*/`
        return 'false';

Step 7 resolve remaining indifferences arbitrarily ;

return 'true' together with the fully refined preference lists $\succ$;

PROOF. The result follows directly from Lemma 5.2 and the definition of $S(a)$. □

LEMMA 5.4. *Let $\succ$ be a strict preference profile that refines I_3. For each applicant a, if $\eta_\mu(a)$ is undefined under $\succ$, then $S(a)$ is empty. Otherwise, if $\eta_\mu(a)$ exists under $\succ$, then $\eta_\mu(a) \in T(a)$.*

PROOF. We first show that if $S(a)$ is not empty then $\eta_\mu(a)$ exists under all $\succ \lhd I_3$. Assume that $S(a)$ is not empty. Therefore there exists an employer e who, under I_3, strictly prefers a to $\mu(e)$ and whom a ranks inferior to $\mu(a)$ under I_4. Hence e qualifies to be $\eta_\mu(a)$, even if no other employer does, for any strict preference profiles $\succ \lhd I_3$.

We now show that if $\eta_\mu(a)$ exists under $\succ \lhd I_3$, then $\eta_\mu(a) \in T(a)$. Assume for contradiction that there exists a strict preference profile $\succ \lhd I_3$ under which $\eta_\mu(a)$ exists and is not in $T(a)$. Let e denote $\eta_\mu(a)$ under $\succ$. As $\eta_\mu(a)$ exists, $S(a)$ is not empty. As $e \notin T(a)$, there must exist an employer $e^* \in S(a)$ whom a prefers to e under I_3 and hence under $\succ$. Also, from the definition of $S(a)$, it follows that $a \succ_{e^*} \mu(e^*)$. Hence, following the definition of an exposed rotation and the fact that $\mu(a) \succ_a e^*$ (see Lemma 5.3), e cannot be $\eta_\mu(a)$ under $\succ$, a contradiction. $\square$

We next identify conditions under which an agent is labeled good.

LEMMA 5.5. *An applicant a is labeled good in the algorithm if and only if one of the following holds for all strict preference profiles $\succ \lhd I_6$: (1) a is unmatched, (2) $\eta_\mu(a)$ is not defined, or (3) $\eta_\mu(a)$ is [already] labeled good.*

PROOF. We first prove that if a is labeled good then at least one of the three conditions must hold. Note that an applicant is labeled good only in Steps 1, 4, and 6. (1) If a is labeled good in Step 1, then she must be unmatched. (2) If a is labeled good in Step 4, the refinements that take place in Step 4 ensure that for all $\succ \lhd I_5$—and hence for all $\succ \lhd I_7$, there is no employer whom a prefers to $\mu(a)$ and who prefers a to his partner. Hence $\eta_\mu(a)$ is not defined. (3) If a is labeled good in Step 6, then it is because there is a good employer e in $T(a)$. The refinements that take place in Step 6 ensure that under $\succ$, (i) e prefers a to $\mu(e)$, and (ii) all employers e' who are ranked between $\mu(a)$ and e prefer $\mu(e')$ to a. To see this, note that as $e \in T(a)$, there is no employer $e^* \in S(a)$—i.e. no employer e^* who strictly prefers a to $\mu(e^*)$ under I_4, such that a strictly prefers e^* to e under I_4—and consequently that the same holds under I_5 and I_6, as no refinement involving a takes place in Steps 5 and 6. Hence, by promoting e ahead of all employers e' such that a is indifferent between e' and e (in Step 6), we ensure that all employers e' who are ranked between $\mu(a)$ and e under $\succ$ weakly prefer $\mu(e)$ to a under I_4 (and hence under I_5 and I_6) and then by promoting a ahead of $\mu(e')$ for such employers e'(in Step 6) it is established that e must be the first employer, after $\mu(a)$, who prefers a to his own partner. Therefore e is $\eta_\mu(a)$ and he is [already] labeled good.

We conclude by showing that if one of the aforementioned conditions holds for an applicant a, then a will be labeled good. (1) If a is unmatched, she is labeled good in Step 1. (2) If $\eta_\mu(a)$ is not defined then $S(a) = \emptyset$ (by Lemma 5.4). Hence a is labeled good in Step 4. (3) If $\eta_\mu(a)$ is labeled good then, as $\eta_\mu(a) \in T(a)$ (by Lemma 5.4), there exists a good employer in $T(a)$. Thus a is labeled good in Step 6. $\square$

LEMMA 5.6. *An employer e is labeled good in the algorithm if and only if one of the following holds: either e is unmatched or $\mu(e)$ is labeled good.*

PROOF. An unmatched employer is labeled good in Step 1. A matched employer is only labeled good in Step 4 or Step 6 along with his partner. $\square$

We are now ready to back up the intuition that good agents do not contribute to exposed rotations.

LEMMA 5.7. *Let I_a denote the SMP instance when applicant a is labeled good. Then a does not contribute to an exposed rotation w.r.t. any strict preference profile $\succ \lhd I_a$.*

PROOF. Assume for contradiction that the claim does not hold. Let a^* be the first applicant who is labeled good in the algorithm for whom the claim does not hold; that is, $\exists \succ \lhd I_{a^*}$ such that a^* belongs to an exposed rotation in μ. As a^* belongs to an exposed rotation, she cannot be unmatched and $\eta_\mu(a)$ exists. Therefore, following Lemma 5.5, $\eta_\mu(a)$ is also labeled good and he must have been labeled good before a^*. Furthermore, following Lemma 5.6, $\mu(\eta_\mu(a))$ was labeled good before a^*. Continuing with this argument we end up with either (i) a path of agents each of whom was labeled good before its successor on the path and the last agent on the path is an applicant a' for whom $\eta_\mu(a')$ is not defined, or (ii) a cycle, not including a^*, where each agent was labeled good before its predecessor on the cycle, or (iii) a cycle ending at a^* where each agent was labeled good before its predecessor on the cycle. Note that case (i) implies that a^* does not belong to an exposed rotation (a contradiction); case (ii) implies that a^* is not the first good applicant who belongs to an exposed rotation (a contradiction); and case (iii) implies that $\mu(a^*)$ was labeled good before a^*. The algorithm labels a matched employer good only when his partner is also labeled good at the same time, hence $\mu(a^*)$ cannot have been labeled good earlier, a contradiction. $\square$

Next we show that if the algorithm returns 'false', then μ admits an exposed rotation under all strict preference profiles that refine the current SMP instance.

LEMMA 5.8. *Assume that Algorithm 4 returns 'false'. Then for all strict preference profiles $\succ \lhd I_6$, there exists a rotation exposed in μ where all agents in the rotation are labeled bad.*

PROOF. The proof follows from the definition of an exposed rotation, Definition 3.13, Lemma 5.5 and Lemma 5.6. Let $\succ \lhd I_6$. Let a be an applicant labeled bad. By Lemma 5.5, $\eta_\mu(a)$ must exist and must be labeled bad. By Lemma 5.6, $\eta_\mu(a)$ is matched under μ and $\mu(\eta_\mu(a))$ is also labeled bad. Continuing with this argument, and as there is a finite number of agents, we have to reach an employer e who is labeled bad and who is matched to a: an exposed rotation where all agents in the rotation are labeled bad. $\square$

We can now prove that IS-EMP-OPT is sound and complete, and terminates in polynomial time.

THEOREM 5.9. *Given an instance I of SMP and a matching μ that is stable w.r.t. I, we can decide in polynomial time whether there exists a strict preference profile $\succ \lhd I$ such that μ is employer-optimal w.r.t. $\succ$.*

PROOF. We first prove that if IS-EMP-OPT returns $\succ$, then μ is the employer-optimal matching of $\succ$. The stability of μ follows from Lemma 5.2. It remains to show that there is no rotation exposed in μ. Note that as the algorithm returns $\succ$ rather than "false", all agents must be labeled good. It then follows from Lemma 5.7 that no applicant contributes to an exposed rotation.

We now prove that if IS-EMP-OPT returns "false", then there is no strict preference profile $\succ$ that refines I where μ is employer-optimal w.r.t. $\succ$. It is enough to show that for all $\succ \lhd I$ such that μ is stable under $\succ$, there exists an exposed rotation in μ. Assume that this is not the case. Hence, there exists a strict preference profile $\succ \lhd I$ under which (i) μ is stable, and hence by Lemma 5.2 $\succ \lhd I_2$, (ii) there is no rotation exposed in μ, and (iii) each applicant a matched in μ ranks $\mu(a)$ ahead of all employers who are incomparable with $\mu(a)$ in a's preference ordering (by Lemma 5.1), and hence $\succ \lhd I_3$. But, by Lemma 5.8, $\succ$ cannot refine I_6. Take an applicant a who is labeled bad when the algorithm stops and returns "false". Note that a must be matched. We first show that $\eta_\mu(a)$ exists under $\succ$. Assume, for a contradiction, that $\eta_\mu(a)$ is undefined. Since $\succ \lhd I_3$, it follows from Lemma 5.4 that $S(a)$ is empty. Hence, by Lemma 5.5, a must be labeled good, and so must be $\mu(a)$ (by Lemma 5.6), a contradiction. So $\eta_\mu(a')$ is defined for all applicants a' who are matched under μ and labeled bad when the algorithm stops. Furthermore, again by Lemma 5.5, $\eta_\mu(a')$ is labeled bad as well. Let a_{i_0} be a bad applicant, hence $\eta_\mu(a_{i_0})$ is defined and is labeled bad. Let e_{i_1} denote $\eta_\mu(a_{i_0})$. As e_{i_1} is labeled bad, so must be $\mu(e_{i_1})$, which we denote by a_{i_1}. Continuing with this argument, as there is a finite number of bad agents, we have to reach an employer $e_{i_{r-1}}$ whose partner is a_0. Hence $(a_{i_0}, e_{i_0}), ..., (a_{i_{r-1}}, e_{i_{r-1}})$ constitute an exposed rotation, a contradiction.

To see that Algorithm 4 runs in time polynomial in the input size, we first note that, under any reasonable assumption on the way the input is represented, the fundamental operations such as deletion, reinstatement, and promotion of an entry in the preference structures, and the identification of the sets $S(a)$ and $T(a)$, can all be carried out in polynomial time. Furthermore, the number of executions of every loop is bounded by a polynomial—in particular, the main loop in Step 6 is executed at most n times. $\square$

6. CONCLUSION

We investigated the problem of reasoning about employer-optimal matchings in settings with partial information. By symmetry, our results also apply to applicant-optimal matchings. A number of open problems remain. (1) In the context of Algorithm IS-EMP-OPT, can we can succinctly characterize and/or efficiently generate the set of all SMI instances that refine the given SMP instance and for which the given stable matching μ is employer-optimal? (2) To what extent can our algorithmic results be generalized to apply to many-to-one matching markets? (3) What can be said (e.g., via empirical studies) about the likelihood that a given partial preference profile admits a pervasive employer-optimal matching?

REFERENCES

GALE, D. AND SHAPLEY, L. 1962. College admission and the stability of marriage. *The American Mathematical Monthly 69,* 1, 9–15.

GUSFIELD, D. AND IRVING, R. W. 1989. *The stable marriage problem: structure and algorithms*. MIT Press, Cambridge, MA, USA.

IRVING, R. AND MANLOVE, D. 2009. Finding large stable matchings. *ACM Journal of Experimental Algorithmics 14,* 2.

IRVING, R. W. 1994. Stable marriage and indifference. *Discrete Applied Mathematics 48,* 3, 261–272.

IRVING, R. W., MANLOVE, D., AND SCOTT, S. 2008. The stable marriage problem with master preference lists. *Discrete Applied Mathematics 156,* 15, 2959–2977.

IRVING, R. W., MANLOVE, D. F., AND SCOTT, S. 2000. The hospitals/residents problem with ties. In *SWAT*. Lecture Notes in Computer Science Series, vol. 1851. 259–271.

IRVING, R. W., MANLOVE, D. F., AND SCOTT, S. 2003. Strong stability in the hospitals/residents problem. In *STACS*, H. Alt and M. Habib, Eds. Lecture Notes in Computer Science Series, vol. 2607. Springer-Verlag GmbH, 439–450.

KNUTH, D. 1976. *Mariages Stables*. Montréal: Les Presses de l'Université de Montréal.

KONCZAK, K. AND LANG, J. 2005. Voting procedures with incomplete preferences. *Multi-disciplinary Workshop on Advances in Preference Handling*.

MANLOVE, D. 1999. Stable marriage with ties and unacceptable partners. Tech. Rep. TR-1999-29, University of Glasgow, Department of Computing Science.

MANLOVE, D. F. 2002. The structure of stable marriage with indifference. *Discrete Applied Mathematics 122,* 13, 167 – 181.

MANLOVE, D. F. 2013. *Algorithmics of matching under preferences*. World Scientific.

MANLOVE, D. F., IRVING, R. W., IWAMA, K., MIYAZAKI, S., AND MORITA, Y. 2002. Hard variants of stable marriage. *Theoretical Computer Science 276,* 1-2, 261 – 279.

RASTEGARI, B., CONDON, A., IMMORLICA, N., AND LEYTON-BROWN, K. 2013. Two-sided matching with partial information. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*. ACM, 733–750.

ROTH, A. 1982. The economics of matching: stability and incentives. *Mathematics of Operations Research 7,* 4, 617–628.

ROTH, A. 1996. The national residency matching program as a labor market. *Journal of the American Medical Association 275,* 13, 1054–1056.

ROTH, A. AND SOTOMAYOR, M. 1990. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press.

SCOTT, S. 2005. A study of stable marriage problems with ties. Ph.D. thesis, Department of Computing Science, University of Glasgow.

XIA, L. AND CONITZER, V. 2011. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research (JAIR) 41*, 25–67.