

Classified Stable Matching

Chien-Chung Huang*

Max-Planck-Institut für Informatik, Saarbrücken, 66123, Germany
villars@mpi-inf.mpg.de

Abstract. We introduce the CLASSIFIED STABLE MATCHING problem, a problem motivated by academic hiring. Suppose that a number of institutes are hiring faculty members from a pool of applicants. Both institutes and applicants have preferences over the other side. An institute classifies the applicants based on their research areas (or any other criterion), and, for each class, it sets a lower bound and an upper bound on the number of applicants it would hire in that class. The objective is to find a stable matching from which no group of participants has reason to deviate. Moreover, the matching should respect the upper/lower bounds of the classes.

In the first part of the paper, we study classified stable matching problems whose classifications belong to a fixed set of “order types.” We show that if the set consists entirely of downward forests, there is a polynomial-time algorithm; otherwise, it is NP-complete to decide the existence of a stable matching.

In the second part, we investigate the problem using a polyhedral approach. Suppose that all classifications are laminar families and there is no lower bound. We propose a set of linear inequalities to describe stable matching polytope and prove that it is integral. This integrality allows us to find various optimal stable matchings using Ellipsoid algorithm. By studying the geometric structure of fractional stable matchings, we are able to generalize a theorem of Teo and Sethuraman in the context of classified stable matching: given any number of stable matchings, if every applicant is assigned to his median choice among all stable matchings, the outcome is still a stable matching. Finally, a ramification of our result is the description of the stable matching polytope for the many-to-many (unclassified) stable matching problem. This answers an open question posed by Sethuraman, Teo and Qian.

* Research supported by an Alexander von Humboldt fellowship

1 Introduction

Imagine that a number of institutes are recruiting faculty members from a pool of applicants. Both sides have their preferences. It would be ideal if there is a matching from which no applicant and institute have reason to deviate. If an applicant prefers another institute to the one he is assigned to (or maybe he is unassigned) and this institute also prefers him to any one of its assigned applicants, then this institute-applicant pair is a *blocking pair*. A matching is *stable* if there is no blocking pair.

The above scenario is the well-studied HOSPITALS/RESIDENTS problem [8, 12] in a different guise. It is known that stable matchings always exist and can be found efficiently by the Gale-Shapley algorithm. However, real world situations can be more complicated. An institute may have its own hiring policy and may find certain sets of applicants *together* unacceptable. For example, an institute may have reasons to avoid hiring too many applicants graduated from the same school; or it may want to diversify its faculty so that it can have researchers in many different fields.

This concern motivates us to consider the following problem. An institute, besides giving its preference among the applicants, also classifies them based on their expertise (or some other criterion). For each class, it sets an upper bound and a lower bound on the number of applicants it would hire. Each institute defines its own classes and classifies the applicants in its own way (and the classes need not be disjoint). We consider this flexibility a desirable feature, as there are some research fields whose boundaries are blurred; moreover, some versatile researchers may be hard to categorize.

We call the above problem CLASSIFIED STABLE MATCHING. Even though motivated by academic hiring, it comes up any time objects on one side of the matching have multiple partners that may be classified. For example, the two sides can be jobs and machines; each machine is assigned several jobs but perhaps cannot take two jobs with heavy memory requirements.

To make the problem precise, we introduce necessary notation and terminology. A set \mathcal{A} of applicants and a set \mathcal{I} of institutes are given. Each applicant/institute has a strictly-ordered (but not necessarily complete) preference list over the other side. The notation \succeq_x indicates either strictly better or equal in terms of preference of an entity $e \in \mathcal{A} \cup \mathcal{I}$ while \succ_e means strictly better. For example, if applicant $a \in \mathcal{A}$ strictly prefers institute $i \in \mathcal{I}$ to another institute $i' \in \mathcal{I}$, we write $i \succ_a i'$. The preference list of institute i is denoted as \mathbb{L}^i . The set of applicants on \mathbb{L}^i who rank higher (respectively lower) than some particular applicant a are written as $\mathbb{L}_{\succ_a}^i$ (respectively $\mathbb{L}_{\prec_a}^i$).

An institute i has a *capacity* $Q(i) \in \mathbb{Z}^+$, the maximum number of applicants it can hire. It defines its own classification $\mathbb{C}(i) = \{C_j^i\}_{j=1}^{|\mathbb{C}(i)|}$, which is a family of sets over the applicants in its preference list. Each class $C_j^i \in \mathbb{C}(i)$ has an upperbound $q^+(C_j^i) \in \mathbb{Z}^+$ and a lowerbound $q^-(C_j^i) \in \mathbb{Z}^+ \cup \{0\}$, on the number of applicants it would hire in that class. Given a matching μ , $\mu(a)$ is the institute applicant a is assigned to. We write $\mu(i) = (a_{i1}, a_{i2}, \dots, a_{ik})$, $k \leq Q(i)$ to denote the set of applicants institute i gets in μ , where a_{ij} are listed in decreasing order based on its preference list. In this paper, we will slightly abuse notation, treating an (ordered) tuple such as $\mu(i)$ as a set.

Definition 1. *Given a tuple $t = (a_{i1}, a_{i2}, \dots, a_{ik})$ where a_{ij} are ordered based on their decreasing rankings on institute i 's preference list, it is said to be a feasible tuple of institute i , or just feasible for short, if the following conditions hold:*

- $k \leq Q(i)$;
- given any class $C_j^i \in \mathbb{C}(i)$, $q^-(C_j^i) \leq |t \cap C_j^i| \leq q^+(C_j^i)$.

Definition 2. *A matching μ is feasible if all the tuples $\mu(i)$, $i \in \mathcal{I}$ are feasible. A feasible matching is stable if and only if there is no blocking group. A blocking group is defined as follows. Let $\mu(i) =$*

$(a_{i1}, a_{i2}, \dots, a_{ik}), k \leq Q(i)$. A feasible tuple $g = (a'_{i1}, a'_{i2}, \dots, a'_{ik'}), k \leq k' \leq Q(i)$, forms a blocking group $(i;g)$ with institute i if

- for $1 \leq j \leq k', i \succeq_{a'_{ij}} \mu(a'_{ij})$, and for $1 \leq j \leq k, a'_{ij} \succeq_i a_{ij}$, and
- either there exists $l, 1 \leq l \leq k$ such that $a'_{il} \succ_i a_{il}$ and $i \succ_{a'_{il}} \mu(a'_{il})$, or that $k' > k$.

Informally speaking, the definition requires that for a blocking group to be formed, all involved applicants have to be willing to switch to, or stay with, institute i . The collection of applicants in the blocking group should still respect the upper and lower bounds in each class; moreover, the institute gets a *strictly better deal* (in the Pareto-optimal sense). Note that when there is no class lower bound, then the stable matching as defined in Definition 2 can be equivalently defined as a feasible matching with the conventional *blocking pairs* (see Lemma 17 in Section 4). When the class lower bound is present, the definition of the blocking groups captures our intuition that an institute should not indiscriminately replace a lower ranking applicant assigned to it with a higher applicant (with whom it forms a blocking pair), otherwise, the outcome for it may not be a feasible one.

In our proofs, we often use the notation $\mu(i)|^a a'$ to denote a tuple formed by replacing $a \in \mu(i)$ with a' . The order of the tuple $\mu(i)|^a a'$ is still based on institute i 's preference list. If we write $\mu(i)|a$, then this new tuple is obtained by adding a into $\mu(i)$ and re-ordered. In a matching μ , if a class C_j^i is fully-booked, i.e. $|\mu(i) \cap C_j^i| = q^+(C_j^i)$, we often refer to such a class as a “bottleneck” class. We also define an “absorption” operation: given a set B of classes, $\mathfrak{R}(B)$ returns the set of classes which are not entirely contained in other classes in B .

Our Results It would be of interest to know how complicated the classifications of the institutes can be while still allowing the problem a polynomial time algorithm. In this work, we study the CLASSIFIED STABLE MATCHING problems whose classifications belong to a fixed set of “order types.” The order type of a classification is the inclusion poset of all non-empty intersections of classes. We introduce necessary definitions to make our statement precise.

Definition 3. The class inclusion poset $P(i) = (\overline{\mathbb{C}}(i), \preceq)$ of an institute i is composed of sets of the elements from $\mathbb{L}^i: \overline{\mathbb{C}}(i) = \{\overline{C} | \overline{C} = C_j^i \cap C_k^i, \text{ where } C_j^i, C_k^i \in \mathbb{C}(i)\}^1$. In $P(i)$, $\overline{C}_j^i \succ \overline{C}_k^i$ if $\overline{C}_j^i \supset \overline{C}_k^i$; and $\overline{C}_j^i \parallel \overline{C}_k^i$ if $\overline{C}_j^i \not\supset \overline{C}_k^i$ and $\overline{C}_k^i \not\supset \overline{C}_j^i$.

Definition 4. Let $\mathbb{P} = \{P_1, P_2, \dots, P_k\}$ be a set of posets. A classified stable matching instance $(\mathcal{A}, \mathcal{I})$ belongs to the group of \mathbb{P} -CLASSIFIED STABLE MATCHING problems if for each poset $P_j \in \mathbb{P}$, there exists an institute $i \in \mathcal{I}$ whose class inclusion poset $P(i)$ is isomorphic to P_j and conversely, every class inclusion poset $P(i)$ is isomorphic to a poset in \mathbb{P} .

We call a poset a *downward forest* if given any element, no two of its successors are incomparable. Our first main result is the following dichotomy theorem.

Theorem 5. Let $\mathbb{P} = \{P_1, P_2, \dots, P_k\}$ be a set of posets. \mathbb{P} -classified stable matching problems can be solved in polynomial time if every poset $P_j \in \mathbb{P}$ is a downward forest; on the other hand, if \mathbb{P} contains a poset P_j which is not a downward forest, the existence of a stable matching is NP-complete.

¹ Note that this definition allows a class to intersect itself, i.e., $\overline{C} = C_j^i \cap C_j^i$. This implies that $\overline{\mathbb{C}}(i) \supseteq \mathbb{C}(i)$.

We remark that if \mathbb{P} is entirely composed of downward forests, then every classification $\mathbb{C}(i)$ must be a *laminar* family². In this case, we call the problem LAMINAR CLASSIFIED STABLE MATCHING (henceforth **LCSM**).

We present an $O(m^2)$ time algorithm for **LCSM**, where m is the total size of all preferences. Our algorithm is extended from the Gale-Shapley algorithm. Though intuitive, its correctness is difficult to argue due to various constraints³. Furthermore, we show that several well-known structural results in the HOSPITALS/RESIDENTS problem can be further generalized in **LCSM**. On the other hand, if some institute i has a classification $\mathbb{C}(i)$ violating laminarity, then \mathbb{P} must contain a poset which has a “V” (where the “bottom” is induced by two intersecting classes in $\mathbb{C}(i)$ which are its parents “on top.”) We will make use of this fact to design a gadget for our NP-complete reduction. In particular, in our reduction, all institutes only use upperbound constraints. Sections 2 and 3 will be devoted to these results.

Our dichotomy theorem implies a certain limit on the freedom of the classifications defined by the institutes. For example, an institute may want to classify the applicants based on two different criteria simultaneously (say by research fields and gender); however, our result implies this may cause the problem to become intractable.

In the second part, we study **LCSM** using a mathematical programming approach. Assume that there is no lower bound on the classes. We extend the set of linear inequalities used by Baiou and Balinski [4] to describe stable matchings and generalize a bin-packing algorithm of Sethuraman, Teo, and Qian [23] to prove that the polytope is integral. The integrality of our polytope allows us to use suitable objective functions to obtain various optimal stable matchings using Ellipsoid algorithm. As our LP has an exponential number of constraints, we also design a separation oracle.

By studying the geometric structure of fractional stable matchings, we are able to generalize a theorem of Teo and Sethuraman [24]: in (one-to-one) stable marriage, given any number of stable matchings, if we assign every man his median choice among all women with whom he is matched in the given set of matchings and we do similarly for women, the outcome is still a stable matching. This theorem has been generalized in the context of HOSPITALS/RESIDENTS problem [6, 14, 23]. We prove that in **LCSM**, this theorem still holds: if we apply this “median choice operation” on all applicants, the outcome is still a stable matching⁴.

A final ramification of our polyhedral result is an answer to an open question posed by Sethuraman, Teo and Qian [23]: how do we describe the stable matching polytope in the classical “unclassified” many-to-many stable matching problem? We show this problem can be reduced to **LCSM** by suitable cloning and classifications.

All the polyhedral results will be presented in Section 4. In Section 5 we conclude. Omitted proofs and details can be found in the appendix.

1.1 Related Work

Stable matching problems have drawn the intensive attention of researchers in various disciplines in the past decades since the seminal paper of Gale and Shapley [8]. For a summary, see [12, 15, 19]. Vande Vate [25] initiated the study of stable matching using mathematical programming approach; further developments using this approach can be found in [2, 4, 18, 20, 22–24].

² A laminar family \mathcal{F} has no pair of intersecting classes, that is, if $A, B \in \mathcal{F}$, then either $A \cap B = \emptyset$, or $A \subseteq B$, or $B \subseteq A$.

³ If there is no lower bound on the classes, the proof can be significantly simplified.

⁴ However, this “median choice operation” on the applicants’ side does not mean that all institutes will get their (lexicographically) median outcome. Thus, the “fairness” of the resultant matching is to a certain degree lost in our generalizations. See Section 4 for details.

Fleiner [7] studied the many-many stable matching in a much more general context. Using a fixed-point approach, he proved that stable matchings always exist provided that the preference of each entity is a *substitutable choice function*. Roughly speaking, such a function can be realized by imposing a matroid over a linear order of elements. In **LCSM**, supposing that there is no lower bound on the classes, then each laminar family is equivalent to a partition matroid. We prove that stable matchings always exist in this situation. Hence, our algorithm in Section 2 can be seen as a constructive proof of a special case of Fleiner’s existence theorem.

Abraham, Irving and Manlove introduced the **STUDENT-PROJECT ALLOCATION** problem [3]. It can be shown that in **LCSM**, if all classifications are just partitions over the applicants and there is no lower bound, our problem is equivalent to a special case of their problem. They posed the open question whether there is a polynomial time algorithm for their problem if there is lower bound on the projects (classes). Our result in Section 2 gives a partial positive answer.

Two recent works [5, 13] also consider the Hospitals/Residents problem in the context of having lower bounds on the hospitals’ side. In [5], an interesting variation of the Hospitals/Residents problem which has a similar flavor to the current work is defined as follows: each hospital has its individual quota and sets of hospitals may also have collective quotas.

2 An Algorithm for Laminar Classified Stable Matching

In this section, we present a polynomial time algorithm to find a stable matching if it exists in the given **LCSM** instance, otherwise, to report that none exists.

We pre-process our instance as follows. If applicant a is on institute i ’s preference list, we add a class $C_{a1}^i = \{a\}$ into $\mathbb{C}(i)$. Furthermore, we also add a class $C_{\#}^i$ into $\mathbb{C}(i)$ including all applicants in \mathbb{L}^i . After this pre-processing, the set of classes in $\mathbb{C}(i)$ form a tree whose root is the $C_{\#}^i$; moreover, an applicant a belongs to a sequence of classes $a(\mathbb{C}(i)) = (C_{a1}^i, C_{a2}^i, \dots, C_{az}^i (= C_{\#}^i))$, which forms a path from the leaf to the root in the tree (i.e., C_{aj}^i is a super class of $C_{aj'}^i$, provided $j' < j$.) For each non-leaf class C_j^i , let $c(C_j^i)$ denote the set of its child classes in the tree. We can assume without loss of generality that $q^-(C_j^i) \geq \sum_{C_k^i \in c(C_j^i)} q^-(C_k^i)$ for any non-leaf class C_j^i . Finally, let $q^+(C_{\#}^i) := Q(i)$, $q^-(C_{\#}^i) := \sum_{C_k^i \in c(C_{\#}^i)} q^-(C_k^i)$; for all applicants $a \in \mathbb{L}^i$, $q^+(C_{a1}^i) := 1$ and $q^-(C_{a1}^i) := 0$.

Our algorithm finds an *applicant-optimal-institute-pessimal* stable matching. The applicant-optimality means that all applicants get the best outcome among all stable matchings; on the other hand, institute-pessimality means that all institutes get an outcome which is “lexicographically” the worst for them. To be precise, suppose that $\mu(i) = (a_{i1}, a_{i2}, \dots, a_{ik})$ and $\mu'(i) = (a_{i1}, a_{i2}, \dots, a_{ik})$ are the outcomes of two stable matchings for institute i ⁵. If there exists $k' \leq k$ so that $a_{ij} = a'_{ij}$, for all $1 \leq j \leq k' - 1$ and $a_{ik'} \succ_i a'_{ik'}$, then institute i is lexicographically better off in μ than in μ' .

We now sketch the high-level idea of our algorithm. We let applicants “propose” to the institutes from the top of their preference lists. Institutes make the decision of acceptance/rejection of the proposals based on certain rules (to be explained shortly). Applicants, if rejected, propose to the next highest-ranking institutes on their lists. The algorithm terminates when all applicants either end up with some institutes, or run out of their lists. Then we check whether the final outcome meets the upper and lower bounds of all classes. If yes, the outcome is a stable matching; if no, there is no stable matching in the given instance.

How the institutes make the acceptance/rejection decisions is the core of our algorithm. Intuitively, when an institute gets a proposal, it should consider two things: (i) will adding this new applicant

⁵ In **LCSM**, an institute always gets the same number of applicants in all stable matchings. See Theorem 15 below.

violate the upper bound of some class? (ii) will adding this applicant deprive other classes of their necessary minimum requirement? If the answer to any of the two questions is positive, the institute should not just take the new applicant unconditionally; instead, it has to reject someone it currently has (not necessarily the new comer).

Below we will design two invariants for all classes of an institute. Suppose that institute i gets a proposal from applicant a , who belongs to a sequence of classes $a(\mathbb{C}(i)) = (C_{a_1}^i, C_{a_2}^i, \dots, C_{\#}^i)$. We check this sequence of classes from the leave to the root. If adding applicant a into class $C_{a_j}^i$ does not violate these invariants, we climb up and see if adding applicant a into $C_{a_{(j+1)}}^i$ violates the invariant. If we can reach all the way to $C_{\#}^i$ without violating the invariants in any class in $a(\mathbb{C}(i))$, applicant a is just added into institute i 's new collection. If, on the other hand, adding applicant a into $C_{a_{(j+1)}}^i$ violates the invariants, institute i rejects *some* applicant in $C_{a_{(j+1)}}^i$ who is from *a sequence of subclasses of $C_{a_{(j+1)}}^i$ which can afford to lose one applicant*.

We define a *deficiency* number $\Delta(C_j^i)$ for each class $C_j^i \in \mathbb{C}(i)$. Intuitively, the deficiency number indicates how many more applicants are necessary for class C_j^i to meet the lower bound of all its subclasses. This intuition translates into the following invariant:

$$\textbf{Invariant A: } \Delta(C_j^i) \geq \sum_{C_k^i \in c(C_j^i)} \Delta(C_k^i), \forall C_j^i \in \mathbb{C}(i), c(C_j^i) \neq \emptyset, \forall i \in \mathcal{I}.$$

In the beginning, $\Delta(C_j^i)$ is set to $q^-(C_j^i)$ and we will explain how $\Delta(C_j^i)$ is updated shortly. Its main purpose is to make sure that after adding some applicants into C_j^i , there is still enough “space” for other applicants to be added into C_j^i so that we can satisfy the lower bound of all subclasses of C_j^i . In particular, we maintain

$$\textbf{Invariant B: } q^-(C_j^i) \leq |\mu(i) \cap C_j^i| + \Delta(C_j^i) \leq q^+(C_j^i), \forall C_j^i \in \mathbb{C}(i), \forall i \in \mathcal{I}.$$

We now explain how $\Delta(C_j^i)$ is updated. Under normal circumstances, we decrease $\Delta(C_j^i)$ by 1 once we add a new applicant into C_j^i . However, if Invariant A is already “tight”, i.e., $\Delta(C_j^i) = \sum_{C_k^i \in c(C_j^i)} \Delta(C_k^i)$, then we add the new applicant C_j^i without decreasing $\Delta(C_j^i)$. The same situation may repeat until the point that $|\mu(i) \cap C_j^i| + \Delta(C_j^i) = q^+(C_j^i)$ and adding another new applicant in C_j^i is about to violate Invariant B. In this case, something has to be done to ensure that Invariant B holds: some applicant in C_j^i has to be rejected, and the question is whom?

Let us call a class a *surplus* class if $|\mu(i) \cap C_j^i| + \Delta(C_j^i) > q^-(C_j^i)$ and we define an *affluent set* for each class C_j^i as follows:

$$\$(C_j^i, \mu(i)) = \{a | a \in \mu(i) \cap C_j^i; \text{ for each } C_{j'}^i \in a(\mathbb{C}(i)) \text{ and } C_{j'}^i \subset C_j^i, |\mu(C_{j'}^i)| + \Delta(C_{j'}^i) > q^-(C_{j'}^i)\}.$$

In words, the affluent set $\$(C_j^i, \mu(i))$ is composed of the set of applicants currently assigned to institute i , part of C_j^i , and each of whom belonging to a sequence of surplus subclasses of C_j^i . In our algorithm, to prevent Invariant B from being violated in a non-leaf class C_j^i , institute i rejects the lowest ranking applicant a in the affluent set $\$(C_j^i, \mu(i))$.

The pseudo-code of the algorithm is presented in Figure 1.

Initialization

0: $\forall i \in \mathcal{I}, \forall C_j^i \in \mathbb{C}(i), \Delta(C_j^i) := q^-(C_j^i);$

Algorithm

1: **While** there exists an applicant a unassigned and he has not been rejected by all institutes on his list
2: Applicant a proposes to the highest ranking institute i to whom he has not proposed so far;
3: Assume that $a(\mathbb{C}(i)) = (C_{a1}^i, C_{a2}^i, \dots, C_{az}^i (= C_{\#}^i));$
4: $\mu(i) := \mu(i) \cup \{a\}$ // Institute i accepts applicant a provisionally;
5: **For** $t = 2$ **To** z // applicant a can be added into C_{a1}^i directly;
6: **If** $\Delta(C_{at}^i) > \sum_{C_k^i \in c(C_{at}^i)} \Delta(C_k^i)$ **Then** $\Delta(C_{at}^i) := \Delta(C_{at}^i) - 1;$
7: **If** $\#(C_{at}^i) + \Delta(C_{at}^i) > q^+(C_{at}^i)$ **Then**
8: Let $\$(C_{at}^i, \mu(i)) = \{a | a \in \mu(i) \cap C_{at}^i; \text{ for each } C_{j'}^i \in a(\mathbb{C}(i)) \text{ and } C_{j'}^i \subset C_{at}^i, |\mu(C_{j'}^i)| + \Delta(C_{j'}^i) > q^-(C_{j'}^i)\};$
9: Let the lowest ranking applicant in $\$(C_{at}^i, \mu(i))$ be $a^\dagger;$
10: $\mu(i) := \mu(i) \setminus \{a^\dagger\}$ // Institute i rejects applicant $a^\dagger;$
11: **GOTO** 1;
12: **If** there exists an institute i with $\Delta(C_{\#}^i) > 0$ **Then** Report "There is no stable matching";
13: **Else** Return the outcome μ , which is a stable matching;

Fig. 1. The pseudo code of the algorithm. It outputs the applicant-optimal-institute-pessimal matching μ if it exists; otherwise, it reports that there is no stable matching.

2.1 Correctness of the Algorithm

In our discussion, C_{at}^i is a class in $a(\mathbb{C}(i))$, where t is the index based on the size of the class C_{at}^i in $a(\mathbb{C}(i))$. Assume that during the execution of the algorithm, applicant a proposes to institute i and when the index t of the **For** loop of Line 5 becomes l and results in a^\dagger being rejected, we say applicant a is *stopped* at class C_{al}^i , and class C_{al}^i *causes* applicant a^\dagger to be rejected.

The first lemma describes some basic behavior of our algorithm.

Lemma 6. (i) *Immediately before the end of the while loop, Invariants A and B hold.*

(ii) *Let applicant a be the new proposer and assume he is stopped at class C_{al}^i . Then*

(iia) *Between the time interval that he makes the new proposal and he is stopped at C_{al}^i , $\Delta(C_{at}^i)$ remains unchanged, for all $1 \leq t \leq l$; moreover, given any class C_{at}^i , $2 \leq t \leq l$, $\Delta(C_{at}^i) = \sum_{C_k^i \in c(C_{at}^i)} \Delta(C_k^i)$.*

(iib) *When a is stopped at a non-leaf class C_{al}^i , $\$(C_{al}^i, \mu(i)) \neq \emptyset$; in particular, any class C_{at}^i , $1 \leq t \leq l-1$, is temporarily a surplus class.*

(iii) *Immediately before the end of the while loop, if class C_j^i is a non-leaf surplus class, then $\Delta(C_j^i) = \sum_{C_k^i \in c(C_j^i)} \Delta(C_k^i)$.*

(iv) *Suppose that applicant a is the new proposer and $C_{al}^i \in a(\mathbb{C}(i))$ causes applicant a^\dagger to be rejected and $a^\dagger(\mathbb{C}(i)) = (C_{a^\dagger 1}^i, C_{a^\dagger 2}^i, \dots, C_{a^\dagger l^\dagger}^i (= C_{al}^i), \dots)$. Then immediately before the end of the while loop, $\Delta(C_{a^\dagger t'}^i) = \sum_{C_k^i \in c(C_{a^\dagger t'}^i)} \Delta(C_k^i)$, for all $2 \leq t' \leq l^\dagger$; moreover, $|\mu(i) \cap C_{a^\dagger t'}^i| + \Delta(C_{a^\dagger t'}^i) = q^+(C_{a^\dagger t'}^i)$.*

Proof. (i) can be proved by induction on the number of proposals institute i gets. For (iia), since Invariant A is maintained, if $\Delta(C_{at}^i)$ is decreased for some class C_{at}^i , $1 \leq t \leq l$, the algorithm will ensure that applicant a would not be stopped in any class, leading to a contradiction. Now by (iia), the set of classes $\{C_{at}^i\}_{t=1}^{l-1}$ are (temporarily) surplus classes when applicant a is stopped at C_{al}^i , so $\$(C_{al}^i, \mu(i)) \neq \emptyset$, establishing (iib). Note that this also guarantees that the proposed algorithm is never "stuck."

(iii) can be proved inductively on the number of proposals that institute i gets. Assuming a is the new proposer, there are two cases: (1) Suppose that applicant a is not stopped in any class. Then a class $C_{at}^i \in a(\mathbb{C}(i))$ can become surplus only if the stated condition holds ; (2) Suppose that applicant a is stopped in some class, which causes a^\dagger to be rejected. Let the smallest class containing both a and a^\dagger be $C_{al'}^i$. Applying (iia) and observing the algorithm, it can be verified that only a class $C_{at}^i \subset C_{al'}^i$ can become a surplus class and for such a class, the stated condition holds.

Finally, for the first part of (iv), let $C_{al'}^i$ denote the smallest class containing both a and a^\dagger . Given a class $C_{a^\dagger t'}^i$, if $C_{al'}^i \subseteq C_{a^\dagger t'}^i \subseteq C_{al}^i$, (iia) gives the proof. If $C_{a^\dagger t'}^i \subset C_{al'}^i$, observe that the former must have been a surplus class right before applicant a made the new proposal. Moreover, before applicant a proposed, (iii) implies that for a non-leaf class $C_{a^\dagger t'}^i \subset C_{al'}^i$, the stated condition regarding the deficiency numbers is true. The last statement of (iv) is by the algorithm and Invariant B . \square

Lemma 7. *Assume that $a^\dagger(\mathbb{C}(i)) = (C_{a^\dagger 1}^i, C_{a^\dagger 2}^i, \dots, C_{a^\dagger l^\ddagger}^i, \dots)$. During the execution of the algorithm, suppose that class $C_{a^\dagger l^\ddagger}^i$ causes applicant a^\dagger to be rejected. In the subsequent execution of the algorithm, assuming that $\mu(i)$ is the assignment of institute i at the end of the while loop, then there exists l^\ddagger , where $l^\ddagger \geq l^\dagger$ such that $|\mu(i) \cap C_{a^\dagger l^\ddagger}^i| + \Delta(C_{a^\dagger l^\ddagger}^i) = q^+(C_{a^\dagger l^\ddagger}^i)$; furthermore, for all $2 \leq t \leq l^\ddagger$, all applicants in $\$(C_{a^\dagger t}^i, \mu(i))$ rank higher than a^\dagger . Moreover, for all $2 \leq t \leq l^\ddagger$, $\Delta(C_{a^\dagger t}^i) = \sum_{C_k^i \in c(C_{a^\dagger t}^i)} \Delta(C_k^i)$.*

Proof. We prove based on the induction on the number of proposals institute i receives after a^\dagger is rejected. The base case is when a^\dagger is just rejected. Let $l^\ddagger = l^\dagger$. Then it is obvious that all applicants in the affluent sets $\$(C_{a^\dagger t}^i, \mu(i))$, $2 \leq t \leq l^\ddagger$, rank higher than a^\dagger and the rest of the lemma holds by Lemma 6(iv).

For the induction step, let a be the new proposer. There are four cases. Except the second case, we let l^\ddagger remain unchanged after a 's proposal.

- Suppose that $a \notin C_{a^\dagger l^\ddagger}^i$ and he does not cause anyone in $C_{a^\dagger l^\ddagger}^i$ to be rejected. Then the proof is trivial.
- Suppose that $a \notin C_{a^\dagger l^\ddagger}^i$ and he is stopped in class C_{al}^i , which causes an applicant $a^* \in C_{a^\dagger l^\ddagger}^i$ to be rejected. a^* must be part of the affluent set $\$(C_{a^\dagger l^\ddagger}^i, \mu(i))$ before a proposed. By induction hypothesis, $a^* \succ_i a^\dagger$. Moreover, since a^* is chosen to be rejected, all the applicants in the (new) affluent sets $\$(C_{a^\dagger t}^i, \mu(i))$, for each class $C_{a^\dagger t}^i$, where $C_{a^\dagger t}^i \subset C_{a^\dagger l^\ddagger}^i \subseteq C_{al}^i$, rank higher than a^* , hence, also higher than a^\dagger . Now let C_{al}^i be the new $C_{a^\dagger l^\ddagger}^i$ and the rest of the lemma follows from Lemma 6(iv).
- Suppose that $a \in C_{a^\dagger l^\ddagger}^i$ and he is not stopped in $C_{a^\dagger l^\ddagger}^i$ or any of its subclasses. We argue that a must be accepted without causing anyone to be rejected; moreover, the applicants in all affluent sets $\$(C_{a^\dagger t}^i, \mu(i))$, for all $1 \leq t \leq l^\ddagger$ remain unchanged. Let the smallest class in $a^\dagger(\mathbb{C}(i))$ containing a be $C_{a^\dagger \tilde{l}}^i$. Note that before a proposed, the induction hypothesis states that $|\mu(i) \cap C_{a^\dagger \tilde{l}}^i| + \Delta(C_{a^\dagger \tilde{l}}^i) = q^+(C_{a^\dagger \tilde{l}}^i)$. As applicant a is not stopped at $C_{a^\dagger \tilde{l}}^i$, the set of values $\Delta(C_{a^\dagger t}^i)$, $\tilde{l} \leq t \leq l^\ddagger$, must have decreased during his proposal and this implies that he will not be stopped in any class. Now let $a(\mathbb{C}(i)) = (C_{a 1}^i, \dots, C_{al}^i, C_{a(l+1)}^i (= C_{a^\dagger \tilde{l}}^i), \dots)$. Since $\Delta(C_{a^\dagger \tilde{l}}^i) = \sum_{C_k^i \in c(C_{a^\dagger \tilde{l}}^i)} \Delta(C_k^i)$ before applicant a proposed by the induction hypothesis, for $\Delta(C_{a^\dagger \tilde{l}}^i)$ to decrease, $\Delta(C_{al}^i)$ must have decreased as well. Choose the smallest class $C_{al^*}^i \subset C_{a^\dagger \tilde{l}}^i$ whose value $\Delta(C_{al^*}^i)$ has decreased during a 's proposal. We claim that $C_{al^*}^i$ must have been a non-surplus class *before and after* applicant a 's proposal. If the claim is true, then all the affluent sets $\$(C_{a^\dagger t}^i, \mu(i))$, for all $1 \leq t \leq l^\ddagger$, remain unchanged after applicant a 's proposal.

It is obvious that $C_{al^*}^i \neq C_{a1}^i$. So assume that $C_{al^*}^i$ is a non-leaf class. Suppose for a contradiction that $C_{al^*}^i$ was a surplus class before a proposed. Lemma 6(iii) implies that $\Delta(C_{a^\dagger l^*}^i) = \sum_{C_k^i \in c(C_{a^\dagger l^*}^i)} \Delta(C_k^i)$ before a proposed. Then for $\Delta(C_{a^\dagger l^*}^i)$ to decrease during a 's proposal, $\Delta(C_{a^\dagger(l^*-1)}^i)$ must have decreased as well. But then this contradicts our choice of $C_{a^\dagger l^*}^i$. So we establish that $C_{al^*}^i$ was not surplus and remains so after a 's proposal.

- Suppose that $a \in C_{a^\dagger \tilde{l}}^i$ and when he reaches a subclass of $C_{a^\dagger \tilde{l}}^i$ or the class itself, the latter causes some applicant a^* to be rejected. To avoid trivialities, assume $a \neq a^*$. Let the smallest class in $a^\dagger(\mathbb{C}(i))$ containing a be $C_{a^\dagger \tilde{l}}^i$ and the smallest class in $a^\dagger(\mathbb{C}(i))$ containing a^* be $C_{a^\dagger l^*}^i$. Below we only argue that the case that $C_{a^\dagger \tilde{l}}^i \subseteq C_{a^\dagger l^*}^i$. The other case that $C_{a^\dagger l^*}^i \subset C_{a^\dagger \tilde{l}}^i$ follows essentially the same argument.

After a 's proposal, observe that only the affluent sets $\$(C_{a^\dagger t}^i, \mu(i))$, $\tilde{l} \leq t < l^*$, can have new members (who are from the child class of $C_{a^\dagger \tilde{l}}^i$ containing a). Without loss of generality, let G be the set of new members added into one of the any above sets.

To complete the proof, we need to show that either $G = \emptyset$ or all members in G rank higher than a^\dagger . If before applicant a proposed, a^* belonged to a sequence of surplus classes $C_{a^* t}^i \subset C_{a^\dagger l^*}^i$, he was also part of the affluent set $\$(C_{a^\dagger l^*}^i, \mu(i))$ or part of $\mu(i) \cap C_{a^\dagger 1}^i$ before a proposed. By induction hypothesis, $a^* \succ_i a^\dagger$. Observing Lemma 6(iib), all applicants in G must rank higher than a^* , hence also than a^\dagger . On the other hand, if a^* belongs to some class $C_{a^* t}^i \subset C_{a^\dagger l^*}^i$ which was not surplus before a proposed, then $C_{a^* \tilde{l}}^i = C_{a^* l^*}^i$ and $C_{a^* t}^i$ must also contain a and remain a non-surplus class after a 's proposal. In this case $G = \emptyset$. \square

The following lemma is an abstraction of several counting arguments that we will use afterwards.

Lemma 8. *Let each class C_j^i be associated with two numbers α_j^i and β_j^i and $q^-(C_j^i) \leq \alpha_j^i, \beta_j^i \leq q^+(C_j^i)$. Given any non-leaf class C_j^i , $\alpha_j^i = \sum_{C_k^i \in c(C_j^i)} \alpha_k^i$ and $\beta_j^i \geq \sum_{C_k^i \in c(C_j^i)} \beta_k^i$; moreover, if $\beta_j^i = \sum_{C_k^i \in c(C_j^i)} \beta_k^i$ then such a non-leaf class C_j^i is said to be tight in β . If $\beta_j^i > q^-(C_j^i)$, then C_j^i has to be tight in β .*

- (i) *Given a non-leaf class $C_{a^\dagger l^\dagger}^i$ with $\alpha_{a^\dagger l^\dagger}^i < \beta_{a^\dagger l^\dagger}^i$, we can find a sequence of classes $C_{a^\dagger t}^i \supset \dots \supset C_{a^\dagger 1}^i$, where $\alpha_{a^\dagger t}^i < \beta_{a^\dagger t}^i$, for $1 \leq t \leq l^\dagger$.*
- (ii) *Given a non-leaf class C_x^i with $\alpha_x^i \leq \beta_x^i$, suppose that there exists a leaf class $C_{a^\dagger \phi 1}^i \subset C_x^i$ such that $\alpha_{a^\dagger \phi 1}^i > \beta_{a^\dagger \phi 1}^i$. Moreover, all classes $C_{a^\dagger \phi t}^i$ are tight in β , where $C_{a^\dagger \phi 1}^i \subseteq C_{a^\dagger \phi t}^i \subseteq C_x^i$, then we can find a class $C_{x'}^i$, where $C_{a^\dagger \phi 1}^i \subset C_{x'}^i \subseteq C_x^i$, $\alpha_{x'}^i \leq \beta_{x'}^i$, and two sequences of classes with the following properties:*
 - (iia) $C_{a^\dagger \phi 1}^i \subset C_{a^\dagger \phi 2}^i \subset \dots \subset C_{a^\dagger \phi l^\phi}^i \subset C_{x'}^i$, where $\alpha_{a^\dagger \phi t}^i > \beta_{a^\dagger \phi t}^i$ for $1 \leq t \leq l^\phi$;
 - (iib) $C_{x'}^i \supset C_{a^\dagger t}^i \supset \dots \supset C_{a^\dagger 1}^i$, where $\alpha_{a^\dagger t}^i < \beta_{a^\dagger t}^i$, for $1 \leq t \leq l^\dagger$.

Proof. For (i), since $q^-(C_{a^\dagger l^\dagger}^i) \leq \alpha_{a^\dagger l^\dagger}^i < \beta_{a^\dagger l^\dagger}^i$, class $C_{a^\dagger l^\dagger}^i$ is tight in β . Therefore, $\sum_{C_k^i \in c(C_{a^\dagger l^\dagger}^i)} \alpha_k^i = \alpha_{a^\dagger l^\dagger}^i < \beta_{a^\dagger l^\dagger}^i = \sum_{C_k^i \in c(C_{a^\dagger l^\dagger}^i)} \beta_k^i$. By counting, there exists a class $C_{a^\dagger(l^\dagger-1)}^i \in c(C_{a^\dagger l^\dagger}^i)$ with $q^-(C_{a^\dagger(l^\dagger-1)}^i) \leq \alpha_{a^\dagger(l^\dagger-1)}^i < \beta_{a^\dagger(l^\dagger-1)}^i$. Repeating the same argument gives us the sequence of classes.

For (ii), let us climb up the tree from $C_{a^\dagger \phi 1}^i$ until we meet a class $C_{x'}^i \subseteq C_x^i$ with $\alpha_{x'}^i \leq \beta_{x'}^i$. This gives us the sequence of classes stated in (iia).

Now since the class $C_{x'}^i$ is tight in β , $\sum_{C_k^i \in c(C_{x'}^i)} \alpha_k^i = \alpha_{x'}^i \leq \beta_{x'}^i = \sum_{C_k^i \in c(C_{x'}^i)} \beta_k^i$. Moreover, as $C_{a^\dagger \phi l^\phi}^i \in c(C_{x'}^i)$ and $\alpha_{a^\dagger \phi l^\phi}^i > \beta_{a^\dagger \phi l^\phi}^i$, by counting, we can find another class $C_{a^\dagger t}^i \in c(C_{x'}^i) \setminus \{C_{a^\dagger \phi l^\phi}^i\}$ such that $\beta_{a^\dagger t}^i > \alpha_{a^\dagger t}^i \geq q^-(C_{a^\dagger t}^i)$. Now applying (i) gives us the sequence of classes in (iib). \square

We say that $(i; a)$ is a *stable pair* if there exists any stable matching in which applicant is assigned to institute i . A stable pair is *by-passed* if institute i rejects applicant a during the execution of our algorithm.

Lemma 9. *During the execution of the algorithm, if an applicant a^ϕ is rejected by institute i , then $(i; a^\phi)$ is not a stable pair.*

Proof. We prove by contradiction. Assume that $(i; a^\phi)$ is the first by-passed stable pair and there exists a stable matching μ^ϕ in which $\mu^\phi(a^\phi) = i$. For each class $C_j^i \in \mathbb{C}(i)$, we associate two numbers $\alpha_j^i := |\mu^\phi(i) \cap C_j^i|$ and $\beta_j^i := |\mu(i) \cap C_j^i| + \Delta(C_j^i)$. Here $\Delta(\cdot)$ s are the values recorded in the algorithm right after a^ϕ is rejected (before the end of the while loop); similarly, $\mu(i)$ is the assignment of i at that point.

It is obvious that $\alpha_{a^\phi 1}^i > \beta_{a^\phi 1}^i$ and the class C_x^i causing a^ϕ to be rejected is not $C_{a^\phi 1}^i$. By Lemma 6(iv), all classes $C_{a^\phi t}^i$ are tight in β , where $C_{a^\phi 1}^i \subset C_{a^\phi t}^i \subseteq C_x^i$. It can be checked all the conditions as stated in Lemma 8(ii) are satisfied. In particular, $\beta_x^i = q^+(C_x^i) \geq \alpha_x^i$; moreover, if $\beta_j^i > q^-(C_j^i)$, C_j^i must be tight (by Lemma 6(iii)).

So, we can find two sequences of classes $\{C_{a^\phi t}^i\}_{t=1}^{l^\phi}$ and $\{C_{a^\dagger t}^i\}_{t=1}^{l^\dagger}$, where $C_{a^\phi l^\phi}^i, C_{a^\dagger l^\dagger}^i \in c(C_{x'}^i)$ and $C_{x'}^i \subseteq C_x^i$, with the following properties:

$$\begin{aligned} q^+(C_{a^\phi t}^i) &\geq |\mu^\phi(i) \cap C_{a^\phi t}^i| > |\mu(i) \cap C_{a^\phi t}^i| + \Delta(C_{a^\phi t}^i) \geq q^-(C_{a^\phi t}^i), \forall t, 1 \leq t \leq l^\phi; \\ q^-(C_{a^\dagger t}^i) &\leq |\mu^\phi(i) \cap C_{a^\dagger t}^i| < |\mu(i) \cap C_{a^\dagger t}^i| + \Delta(C_{a^\dagger t}^i) \leq q^+(C_{a^\dagger t}^i), \forall t, 1 \leq t \leq l^\dagger. \end{aligned}$$

The second set of inequalities implies that the classes $\{C_{a^\dagger t}^i\}_{t=1}^{l^\dagger}$ are surplus in μ . Thus there exists an applicant $a^\dagger \in (\mu(i) \setminus \mu^\phi(i)) \cap C_{a^\dagger 1}^i$. Since $(i; a^\phi)$ is the first by-passed stable pair, $i \succ_{a^\dagger} \mu^\phi(a^\dagger)$ and since a^ϕ is rejected instead of a^\dagger , $a^\dagger \succ_i a^\phi$. Now observe the tuple $\mu^\phi(i)|^{a^\phi} a^\dagger$ is feasible due to the above two sets of strict inequalities. Thus we have a group $(i; \mu^\phi(i)|^{a^\phi} a^\dagger)$ to block μ^ϕ , a contradiction. \square

Lemma 10. *At the termination of the algorithm, if there exists an institute $i \in \mathcal{I}$ such that $\Delta(C_{\#}^i) > 0$, there is no stable matching in the given instance.*

Proof. Suppose, for a contradiction, that there exists an institute i with $\Delta(C_{\#}^i) > 0$ and there is a stable matching μ^ϕ . Let μ be the assignment when the algorithm terminates. By Lemma 9, if an applicant is unmatched in μ , he cannot be assigned in μ^ϕ either. So $|\mu^\phi| \leq |\mu|$. In the following, $\Delta(\cdot)$ s refer to values recorded in the final outcome of the algorithm. Consider two cases.

- Suppose that $|\mu^\phi(i)| > |\mu(i) \cap C_{\#}^i|$. Then as $|\mu^\phi| \leq |\mu|$, we can find another institute $i' \neq i$ such that $|\mu^\phi(i')| < |\mu(i') \cap C_{\#}^{i'}|$. For each class $C_j^{i'} \in \mathbb{C}(i')$, let $\alpha_j^{i'} := |\mu^\phi(i') \cap C_j^{i'}|$ and $\beta_j^{i'} := |\mu(i') \cap C_j^{i'}| + \Delta(C_j^{i'})$. It can be checked that the condition stated in Lemma 8(i) is satisfied (note that those $\beta_j^{i'}$ fulfill the condition due to Lemma 6(iii)). Therefore, we can find a sequence of classes $\{C_{a^\dagger t}^{i'}\}_{t=1}^{l^\dagger}$, where $C_{a^\dagger l^\dagger}^{i'} = C_{\#}^{i'}$, and

$$|\mu^\phi(i') \cap C_{a^\dagger t}^{i'}| < |\mu(i') \cap C_{a^\dagger t}^{i'}| + \Delta(C_{a^\dagger t}^{i'}) \leq q^+(C_{a^\dagger t}^{i'}), \forall t, 1 \leq t \leq l^\dagger,$$

where the second inequality follows from Invariant B . Then there exists an applicant $a^\dagger \in (\mu(i') \setminus \mu^\phi(i')) \cap C_{a^\dagger 1}^{i'}$. By Lemma 9, $i' \succ_{a^\dagger} \mu^\phi(a^\dagger)$, giving us a group $(i'; \mu^\phi(i')|^{a^\dagger})$ to block μ^ϕ , a contradiction. Note the feasibility of $\mu^\phi(i')|^{a^\dagger}$ is due to the above set of strict inequalities.

- Suppose that $|\mu^\phi(i)| \leq |\mu(i) \cap C_{\sharp}^i|$. We first claim that C_{\sharp}^i must be a surplus class in $\mu(i)$. If not, then $q^-(C_{\sharp}^i) = \Delta(C_{\sharp}^i) + |\mu(i) \cap C_{\sharp}^i| > |\mu(i) \cap C_{\sharp}^i|$, implying that $|\mu^\phi(i)| \geq q^-(C_{\sharp}^i) > |\mu(i) \cap C_{\sharp}^i|$, a contradiction. So C_{\sharp}^i is a surplus class, and by Lemma 6(iii),

$$|\mu^\phi(i)| = \sum_{C_k^i \in c(C_{\sharp}^i)} |\mu^\phi(i) \cap C_k^i| \leq |\mu(i) \cap C_{\sharp}^i| < |\mu(i) \cap C_{\sharp}^i| + \Delta(C_{\sharp}^i) = \sum_{C_k^i \in c(C_{\sharp}^i)} |\mu(i) \cap C_k^i| + \Delta(C_k^i).$$

For each class $C_j^i \in \mathbb{C}(i)$, let $\alpha_j^i := |\mu^\phi(i) \cap C_j^i|$ and $\beta_j^i := |\mu(i) \cap C_j^i| + \Delta(C_j^i)$ and invoke Lemma 8(i). The above inequality implies that $\alpha_{\sharp}^i < \beta_{\sharp}^i$ and note that by Lemma 6(iii), the condition regarding β is satisfied. Thus we have a sequence of surplus classes $C_{a^\dagger l^\dagger}^i (= C_{\sharp}^i) \supset \dots \supset C_{a^\dagger 1}^i$ so that

$$q^-(C_{a^\dagger t}^i) \leq |\mu^\phi(i) \cap C_{a^\dagger t}^i| < |\mu(i) \cap C_{a^\dagger t}^i| + \Delta(C_{a^\dagger t}^i) \leq q^+(C_{a^\dagger t}^i), \forall t, 1 \leq t \leq l^\dagger,$$

implying that there exists an applicant $a^\dagger \in (\mu(i) \setminus \mu^\phi(i)) \cap C_{a^\dagger 1}^i$ and $i \succ_{a^\dagger} \mu^\phi(a^\dagger)$ by virtue of Lemma 9. The tuple $\mu^\phi(i)|a^\dagger$ is feasible because of the above set of strict inequalities. Now $(i; \mu^\phi(i)|a^\phi)$ blocks μ^ϕ , a contradiction. \square

Lemma 11. *Suppose that in the final outcome μ , for each institute $i \in \mathcal{I}$, $\Delta(C_{\sharp}^i) = 0$. Then μ is a stable matching.*

Proof. For a contradiction, assume that a group $(i; g)$ blocks μ . Let a^ϕ to be the highest ranking applicant in $g \setminus \mu(i)$. Since a^ϕ is part of the blocking group, he must have proposed to and been rejected by institute i during the execution of the algorithm, thus $i \succ_{a^\phi} \mu(a^\phi)$. By Lemma 7, there exists a class $C_{a^\phi l^\dagger}^i$ such that $|\mu(i) \cap C_{a^\phi l^\dagger}^i| + \Delta(C_{a^\phi l^\dagger}^i) = |\mu(i) \cap C_{a^\phi l^\dagger}^i| = q^+(C_{a^\phi l^\dagger}^i)$. Moreover, it is obvious that $|g \cap C_{a^\phi 1}^i| > |\mu(i) \cap C_{a^\phi 1}^i|$. We now make use of Lemma 8(ii) by letting $\alpha_j^i := |g \cap C_j^i|$ and $\beta_j^i := |\mu(i) \cap C_j^i|$ for each class $C_j^i \in \mathbb{C}(i)$. Note that all classes are tight in β , $C_{a^\phi 1}^i \subset C_{a^\phi l^\dagger}^i$, and $|\mu(i) \cap C_{a^\phi l^\dagger}^i| = q^+(C_{a^\phi l^\dagger}^i) \geq |g \cap C_{a^\phi l^\dagger}^i|$, satisfying all the necessary conditions. Thus, we can discover a sequence of classes $\{C_{a^\dagger t}^i\}_{t=1}^{l^\dagger}$ stated in Lemma 8(iib), where $C_{a^\dagger t}^i \in c(C_{a^\phi l}^i)$ and $C_{a^\phi 1}^i \subset C_{a^\phi l}^i \subseteq C_{a^\phi l^\dagger}^i$, such that

$$q^-(C_{a^\dagger t}^i) \leq |g \cap C_{a^\dagger t}^i| < |\mu(i) \cap C_{a^\dagger t}^i| \leq q^+(C_{a^\dagger t}^i), \forall t, 1 \leq t \leq l^\dagger,$$

and there exists an applicant $a^\dagger \in (\mu(i) \setminus g) \cap C_{a^\dagger 1}^i$. The above set of strict inequalities mean that all classes $C_{a^\dagger t}^i$, $1 \leq t \leq l^\dagger$, are surplus classes in μ . Then a^\dagger forms part of the affluent set $\$(C_{a^\phi l}^i, \mu(i))$. By Lemma 7, they all rank higher than a^ϕ . This contradicts our assumption that a^ϕ is the highest-ranking applicant in $g \setminus \mu(i)$. \square

Lemma 12. *Suppose that in the final outcome μ , for each institute $i \in \mathcal{I}$, $\Delta(C_{\sharp}^i) = 0$. Then μ is an institute-pessimal stable matching.*

Proof. Suppose, for a contradiction, that there exists a stable matching μ^ϕ such that there exists an institute i which is lexicographically better off in μ than in μ^ϕ . Let a^\dagger be the highest ranking applicant in $\mu(i) \setminus \mu^\phi(i)$. By Lemma 9, $i \succ_{a^\dagger} \mu^\phi(i)$. If $|\mu^\phi(i) \cap C_{a^\dagger t}^i| < |\mu(i) \cap C_{a^\dagger t}^i| \leq q^+(C_{a^\dagger t}^i)$, for all classes $C_{a^\dagger t}^i \in a^\dagger(\mathbb{C}(i))$, then $(i; \mu^\phi(i)|a^\phi)$ blocks μ^ϕ , a contradiction. So choose the smallest class $C_x^i \in a^\dagger(\mathbb{C}(i))$ such that $|\mu^\phi(i) \cap C_x^i| \geq |\mu(i) \cap C_x^i|$. It is clear that $C_x^i \supset C_{a^\dagger 1}^i$.

Now we apply Lemma 8(ii) by letting $\alpha_j^i := |\mu(i) \cap C_j^i|$ and $\beta_j^i := |\mu^\phi(i) \cap C_j^i|$ for each class $C_j^i \in \mathbb{C}(i)$. It can be checked all conditions stated in Lemma 8(ii) are satisfied. So there exists a class

$C_{x'}^i$ such that $C_{a^\dagger 1}^i \subset C_{x'}^i \subseteq C_x^i$ and we can find two sequences of classes $\{C_{a^\dagger t}^i\}_{t=1}^{l^\dagger}$ and $\{C_{a^\dagger t}^i\}_{t=1}^{l^\dagger}$, where $C_{a^\dagger l^\dagger}^i, C_{a^\dagger 1}^i \in c(C_{x'}^i)$, with the following properties:

$$\begin{aligned} q^+(C_{a^\dagger t}^i) &\geq |\mu(i) \cap C_{a^\dagger t}^i| > |\mu^\phi(i) \cap C_{a^\dagger t}^i| \geq q^-(C_{a^\dagger t}^i), \forall t, 1 \leq t \leq l^\dagger; \\ q^-(C_{a^\dagger t}^i) &\leq |\mu(i) \cap C_{a^\dagger t}^i| < |\mu^\phi(i) \cap C_{a^\dagger t}^i| \leq q^+(C_{a^\dagger t}^i), \forall t, 1 \leq t \leq l^\dagger. \end{aligned}$$

The second set of inequalities implies that we can find an applicant $a^\phi \in (\mu^\phi(i) \setminus \mu(i)) \cap C_{a^\dagger 1}^i$. Recall that we choose a^\dagger to be the highest ranking applicant in $\mu(i) \setminus \mu^\phi(i)$, so $a^\dagger \succ_i a^\phi$. Now we have a group $(i; \mu^\phi(i) |^{a^\phi} a^\dagger)$ to block μ^ϕ to get a contradiction. The feasibility of $\mu^\phi(i) |^{a^\phi} a^\dagger$ is due to the above two sets of strict inequalities. \square

Based on Lemmas 9, 10, 11, and 12, we can draw the conclusion in this section.

Theorem 13. *In $O(m^2)$ time, where m is the total size of all preferences, the proposed algorithm discovers the applicant-optimal-institute-pessimal stable matching if stable matchings exist in the given **LCSM** instance; otherwise, it correctly reports that there is no stable matching. Moreover, if there is no lower bound on the classes, there always exists a stable matching.*

To see the complexity, first note that there can be only $O(m)$ proposals. The critical thing in the implementation of our algorithm is to find out the lowest ranking applicant in each affluent set efficiently. This can be done by remembering the lowest ranking applicant in each class and this information can be updated in each proposal in $O(m)$ time, since the number of classes of each institute is $O(m)$, given that the classes form a laminar family.

2.2 Structures of Laminar Classified Stable Matching

Recall that we define the ‘‘absorption’’ operation as follows. Given a family of classes B , $\mathfrak{R}(B)$ returns the set of classes which are not entirely contained in other classes in B . Note that in **LCSM**, $\mathfrak{R}(B)$ will be composed of a pairwise disjoint set of classes.

We review the well-known rural hospitals theorem [9, 16].

Theorem 14. *(Rural Hospitals Theorem) In the HOSPITALS/RESIDENTS problem, the following holds.*

- (i) *A hospital gets the same number of residents in all stable matchings, and as a result, all stable matchings are of the same cardinality.*
- (ii) *A resident who is assigned in one stable matching gets assigned in all other stable matchings; conversely, an unassigned resident in a stable matching remains unassigned in all other stable matchings.*
- (iii) *An under-subscribed hospital gets the same set of residents in all other stable matchings.*

It turns out that rural hospitals theorem can be generalized in **LCSM**. On the other hand, if some institutes use intersecting classes in their classifications, rural hospitals theorem fails (stable matching size may differ). See the appendix for such an example.

Theorem 15. *(Generalized Rural Hospitals Theorem in LCSM) Let μ be a stable matching. Given any institute i , suppose that B is the set of bottleneck classes in $\mu(i)$ and D is the subset of classes in $\mathbb{C}(i)$ such that $\mathfrak{R}(B) \cup D$ partitions \mathbb{L}^i . The following holds.*

- (i) An institute gets the same number of applicants in all stable matchings, and as a result, all stable matchings are of the same cardinality.
- (ii) An applicant who is assigned in one stable matching gets assigned in all other stable matchings; conversely, an unassigned applicant in a stable matching remains unassigned in all other stable matchings.
- (iii) Every class $C_k^i \in \mathfrak{R}(B) \cup D$ has the same number of applicants in all stable matchings.
- (iv) In a class $C_k^i \subseteq C \in D$, or in a class C_k^i which contains only classes in D , the same set of applicant in class C_k^i will be assigned to institute i in all stable matchings.
- (v) A class C_k^i can have different sets of applicants in different stable matchings only if $C_k^i \subseteq C \in \mathfrak{R}(B)$ or $C_k^i \supseteq C \in \mathfrak{R}(B)$.

Proof. We choose μ^\dagger to be the applicant-optimal stable matching.

Claim A: Suppose that $a \in \mu^\dagger(i) \setminus \mu(i)$. Then there exists a class $C_{al}^i \in a(\mathbb{C}(i))$ such that (i) $|\mu(i) \cap C_{al}^i| = q^+(C_{al}^i)$, and (ii) $a \in C_{al}^i \subseteq C \in \mathfrak{R}(B)$.

Proof of Claim A. If for all classes $C_{al}^i \in a(\mathbb{C}(i))$, $|\mu(i) \cap C_{al}^i| < q^+(C_{al}^i)$, then as μ^\dagger is applicant-optimal, $i \succ_a \mu(a)$, so $(i; \mu(i)|a)$ blocks μ , a contradiction. This establishes (i). (ii) follows easily. \square

Let $\tilde{B} \subseteq B$ be the subset of these bottleneck classes containing at least one applicant $\mu^\dagger(i) \setminus \mu(i)$. By Claim A(ii), $\mathfrak{R}(\tilde{B}) \subseteq \mathfrak{R}(B)$. This implies that for all classes $C_k^i \in (\mathfrak{R}(B) \setminus \mathfrak{R}(\tilde{B})) \cup D$, $|\mu(i) \cap C_k^i| \geq |\mu^\dagger(i) \cap C_k^i|$. Combining this fact with Claim A(ii), we have

$$\begin{aligned}
|\mu(i)| &= \sum_{C_k^i \in (\mathfrak{R}(B) \setminus \mathfrak{R}(\tilde{B})) \cup D} |\mu(i) \cap C_k^i| + \sum_{C_k^i \in \mathfrak{R}(\tilde{B})} |\mu(i) \cap C_k^i| \\
&\geq \sum_{C_k^i \in (\mathfrak{R}(B) \setminus \mathfrak{R}(\tilde{B})) \cup D} |\mu^\dagger(i) \cap C_k^i| + \sum_{C_k^i \in \mathfrak{R}(\tilde{B})} q^+(C_{al}^i) \quad (*) \\
&= \sum_{C_k^i \in (\mathfrak{R}(B) \setminus \mathfrak{R}(\tilde{B})) \cup D} |\mu^\dagger(i) \cap C_k^i| + \sum_{C_k^i \in \mathfrak{R}(\tilde{B})} |\mu^\dagger(i) \cap C_k^i| \\
&= |\mu^\dagger(i)|.
\end{aligned}$$

Thus, $|\mu| \geq |\mu^\dagger|$ and it cannot happen that $|\mu| > |\mu^\dagger|$, otherwise, there exists an applicant who is assigned in μ but not in μ^\dagger . This contradicts the assumption that the latter is applicant-optimal. This completes the proof of (i) and (ii) of the theorem.

Since $|\mu| = |\mu^\dagger|$, Inequality (*) holds with equality. We make two observations here.

Observation 1: For each class $C_k^i \in \mathfrak{R}(B)$, it is also a bottleneck in $\mu^\dagger(i)$.

Observation 2: an applicant $a \in \mu^\dagger(i) \setminus \mu(i)$ must belong to a bottleneck class in $\mu^\dagger(i)$.

Let B^\dagger be the set of bottleneck classes in $\mu^\dagger(i)$ and choose D^\dagger so that $\mathfrak{R}(B^\dagger) \cup D^\dagger$ partitions \mathbb{L}^i . By Observation 2, each applicant in $\mu^\dagger(i) \cap C_k^i$, where $C_k^i \in D^\dagger$, must be part of $\mu(i)$. So for each class $C_k^i \in D^\dagger$, $|\mu(i) \cap C_k^i| \geq |\mu^\dagger(i) \cap C_k^i|$. We claim that it cannot happen that $|\mu(i) \cap C_k^i| > |\mu^\dagger(i) \cap C_k^i|$. Suppose not. Then by (i), there are two possible cases.

- There exists another class $C_{k'}^i \in D^\dagger$ so that $|\mu(i) \cap C_{k'}^i| < |\mu^\dagger(i) \cap C_{k'}^i|$. Then we have a contradiction to Observation 2.

- There exists another class $C_{k'}^i \in \mathfrak{R}(B^\dagger)$ so that $|\mu(i) \cap C_{k'}^i| < |\mu^\dagger(i) \cap C_{k'}^i|$. For each class $C_j^i \in \mathbb{C}(i)$, let $\alpha_j^i := |\mu(i) \cap C_j^i|$ and $\beta_j^i := |\mu^\dagger(i) \cap C_j^i|$. Then we can invoke Lemma 8(i) and find an applicant $a^\phi \in \mu^\dagger(i) \setminus \mu(i)$ so that for each class $C_{a^\phi t}^i \in a^\phi(\mathbb{C}(i))$, $C_{a^\phi t}^i \subseteq C_{k'}^i$, $|\mu(i) \cap C_{a^\phi t}^i| < |\mu^\dagger(i) \cap C_{a^\phi t}^i| \leq q^+(C_{a^\phi t}^i)$. Then by Claim A(ii) and Observation 1, there must exist another class $C_{k''}^i \in \mathfrak{R}(B)$ containing a^ϕ and $C_{k''}^i \supset C_{k'}^i$. By Observation 1, $C_{k''}^i$ is also a bottleneck class in $\mu^\dagger(i)$. This contradicts the assumption that $C_{k'}^i \in \mathfrak{R}(B^\dagger)$.

So we have that for each class $C_k^i \in D^\dagger$, $|\mu(i) \cap C_k^i| = |\mu^\dagger(i) \cap C_k^i|$. For each class $C_k^i \in B^\dagger$, we can use the same argument to show that $|\mu(i) \cap C_k^i| = |\mu^\dagger(i) \cap C_k^i|$. This gives us (iii) and (iv). (v) is a consequence of (iv). \square

3 NP-completeness of \mathbb{P} -Classified Stable Matching

Theorem 16. *Suppose that the set of posets $\mathbb{P} = \{P_1, P_2, \dots, P_k\}$ contains a poset which is not a downward forest. Then it is NP-complete to decide the existence of a stable matching in \mathbb{P} -classified stable matching. This NP-completeness holds even if there is no lower bound on the classes.*

Our reduction is from ONE-IN-THREE SAT. It is involved and technical, so we just highlight the idea here. As \mathbb{P} must contain a poset that has a “V” in it, some institutes use intersecting classes. In this case, even if there is no lower bound on the classes, it is possible that the given instance disallows any stable matching. We make use of this fact to design a special gadget. The main technical difficulty of our reduction lies in that in the most strict case, we can use at most two classes in each institute’s classification.

4 Polyhedral Approach

In this section, we take a polyhedral approach to studying **LCSM**. We make the simplifying assumption that there is no lower bound. In this scenario, we can use a simpler definition to define a stable matching.

Lemma 17. *In **LCSM**, if there is no lower bound, i.e., given any class C_j^i , $q^-(C_j^i) = 0$, then a stable matching as defined in Definition 2 can be equivalently defined as follows. A feasible matching μ is stable if and only if there is no blocking pair. A pair (i, a) is blocking, given that $\mu(i) = (a_{i1}, a_{i2}, \dots, a_{ik})$, $k \leq Q(i)$, if*

- $i \succ_a \mu(a)$;
- for any class $C_{at}^i \in a(\mathbb{C}(i))$, $|\mathbb{L}_{\succ_a}^i \cap \mu(i) \cap C_{at}^i| < q^+(C_{at}^i)$.

The definition of blocking pairs suggests a generalization of the *comb* used by Baiou and Balinski [4].

Definition 18. *Let $\Gamma = \mathcal{I} \times \mathcal{A}$ denote the set of acceptable institute-applicant pairs. The shaft $S(A^i)$, based on a feasible tuple A^i of institute i , is defined as follows:*

$$S(A^i) = \{(i, a') \in \Gamma : \forall C_j^i \in a'(\mathbb{C}(i)), |\mathbb{L}_{\succ_{a'}}^i \cap A^i \cap C_j^i| < q^+(C_j^i)\}.$$

The tooth $T(i, a)$ is defined for every $(i, a) \in \Gamma$ as follows:

$$T(i, a) = \{(i', a) \in \Gamma : i' \succeq_a i\}.$$

In words, (i, a') forms part of the shaft $S(A^i)$, only if the collection of a' and all applicants in A^i ranking strictly higher than a' does not violate the quota of any class in $a'(\mathbb{C}(i))$. We often refer to an applicant $a \in A^i$ as a *tooth-applicant*.

We associate a $|\Gamma|$ -vector x^μ (or simply x when the context is clear) with a matching μ : $x_{ia}^\mu = 1$ if $\mu(a) = i$, otherwise, $x_{ia}^\mu = 0$. Suppose that $\hat{\Gamma} \subseteq \Gamma$. Then $x(\hat{\Gamma}) = \sum_{(i,a) \in \hat{\Gamma}} x_{ia}$. We define a *comb* $K(i, S(A^i))$ as the union of the teeth $\{T(i, a_i)\}_{a_i \in A^i}$ and the shaft $S(A^i)$.

Lemma 19. *Every stable matching solution x satisfies the comb inequality for any comb $K(i, S(A^i))$:*

$$x(K(i, S(A^i))) \equiv x(S(A^i)) + \sum_{a_j \in A^i} x(T(i, a_j) \setminus \{i, a_j\}) \geq |A^i|.$$

It takes a somehow involved counting argument to prove this lemma. Here is the intuition about why the comb inequality captures the stability condition of a matching. The value of the tooth $x(T(i, a))$ reflects the “happiness” of the applicant $a \in A^i$. If $x(T(i, a)) = 0$, applicant a has reason to shift to institute i ; on the other hand, the values collected from the shaft $x(S(A^i))$ indicates the “happiness” of institute i : whether it is getting enough high ranking applicants (of the “right” class). An overall small comb value $x(K(i, S(A^i)))$ thus expresses the likelihood of a blocking group including i and some of the applicants in A^i .

Now let \mathbb{K}^i denote the set all combs of institute i . We write down the linear program:

$$\sum_{i:(i,a) \in \Gamma} x_{ia} \leq 1, \forall a \in \mathcal{A} \quad (1)$$

$$\sum_{a:(i,a) \in \Gamma, a \in C_j^i} x_{ia} \leq q^+(C_j^i), \forall i \in \mathcal{I}, \forall C_j^i \in \mathbb{C}(i) \quad (2)$$

$$x(K(i, S(A^i))) = \sum_{(i,a) \in K(i, S(A^i))} x_{ia} \geq |A^i|, \forall K(i, S(A^i)) \in \mathbb{K}^i, \forall i \in \mathcal{I} \quad (3)$$

$$x_{ia} \geq 0, \forall (i, a) \in \Gamma \quad (4)$$

Suppose there is no classification, i.e., HOSPITALS/RESIDENTS problem. Then this LP reduces to the one formulated by Baiou and Balinski [4]. However, it turns out that this polytope is *not* integral. The example in Figure 2 demonstrates the non-integrality of the polytope. In particular, observe that since μ is applicant-optimal, in all other stable matchings, applicant a_3 can only be matched to i_5 . However, the value $x_{i_1 a_3} = 0.2 > 0$ indicates that x is *outside* of the convex hull of integral stable matchings.

Here we make a critical observation. Suppose that in a certain matching μ^ϕ , applicant a_3 is assigned to i_1 . Then a_2 cannot be assigned to i_1 due to the bound $q^+(C_1^1)$ (see Constraint (2)). If μ^ϕ is to be stable, then a_2 must be assigned to some institute ranking higher than i_1 on his list (in this example there is none), otherwise, $(i, \mu^\phi(i_1)|^{a_3 a_2})$ is bound to be a blocking group in μ^ϕ . Thus, the required constraint to avoid this particular counter-example can be written as

$$x(T(i_1, a_2) \setminus \{i_1, a_2\}) \geq x_{i_1 a_3}.$$

We now formalize the above observation. Given any class $C_j^i \in \mathbb{C}(i)$, we define a *class-tuple* $t_j^i = (a_{i_1}, a_{i_2}, \dots, a_{i_{q^+(C_j^i)}})$. Such a tuple fulfills the following two conditions:

Institute Preferences	Classifications	Class bounds
$i_1:a_1a_6a_7a_2a_3$	$C_1^1 = \{a_2, a_3\}$	$Q(i_1) = 2, q^+(C_1^1) = 1$
$i_2:a_4a_7$		$Q(i_2) = 1$
$i_3:a_2a_4$		$Q(i_3) = 1$
$i_4:a_5a_6$		$Q(i_4) = 1$
$i_5:a_3a_5a_7a_1$	$C_1^5 = \{a_3, a_5\}$	$Q(i_5) = 2, q^+(C_1^5) = 1$

Applicant Preferences

$a_1:i_5i_1$
 $a_2:i_1i_3$
 $a_3:i_1i_5$
 $a_4:i_3i_2$
 $a_5:i_5i_4$
 $a_6:i_4i_1$
 $a_7:i_2i_1i_5$

(Applicant-optimal stable matching) $\mu = \{(i_1; a_2, a_6), (i_2; a_7), (i_3; a_4), (i_4; a_5), (i_5; a_1, a_3)\}$

(A fractional matching x that is *not* inside the convex hull of integral stable matchings)

$x_{i_1a_1} = 0.5, x_{i_1a_6} = 0.8, x_{i_1a_7} = 0.2, x_{i_1a_2} = 0.3, x_{i_1a_3} = 0.2.$
 $x_{i_2a_4} = 0.7, x_{i_2a_7} = 0.3.$
 $x_{i_3a_2} = 0.7, x_{i_3a_4} = 0.3.$
 $x_{i_4a_5} = 0.8, x_{i_4a_6} = 0.2.$
 $x_{i_5a_3} = 0.8, x_{i_5a_5} = 0.2, x_{i_5a_7} = 0.5, x_{i_5a_1} = 0.5.$

Fig. 2. An example showing that the polytope formed by Constraints (1)-(4) is not integral. Since μ is applicant-optimal, in all other stable matchings, applicant a_3 can only be matched to i_5 . However, the value $x_{i_1a_3} = 0.2 > 0$ indicates that x is *outside* of the convex hull of integral stable matchings.

1. $t_j^i \subseteq C_j^i$;
2. if C_j^i is a non-leaf class, then given any subclass C_k^i of C_j^i , $|t_j^i \cap C_k^i| \leq q^+(C_k^i)$.

Let $\mathbb{L}_{\prec t_j^i}^i$ denote the set of applicants ranking lower than all applicants in t_j^i and $\mathbb{L}_{\succeq t_j^i}^i$ the set of applicants ranking at least as high as the lowest ranking applicant in t_j^i .

Lemma 20. *Every stable matching solution x satisfies the following inequality for any class-tuple t_j^i :*

$$\sum_{a_{ij} \in t_j^i} x(T(i, a_{ij}) \setminus \{i, a_{ij}\}) \geq \sum_{a \in C_j^i \cap \mathbb{L}_{\prec t_j^i}^i} x_{ia}.$$

As before, it takes a somehow involved counting argument to prove the lemma but its basic idea is already portrayed in the above example. Now let \mathbb{T}_j^i denote the set of class-tuples in class $C_j^i \in \mathbb{C}(i)$ and $\mathbb{L}_{\prec t_j^i}^i$ denote the set of applicants ranking lower than all applicants in t_j^i . We add the following sets of constraints.

$$\sum_{a_{ij} \in t_j^i} x(T(i, a_{ij}) \setminus \{i, a_{ij}\}) \geq \sum_{a \in C_j^i \cap \mathbb{L}_{\prec t_j^i}^i} x_{ia}, \forall t_j^i \in \mathbb{T}_j^i, \forall \mathbb{T}_j^i \quad (5)$$

Let P_{fsm} denote the set of all solutions satisfying (1)-(5) and P_{sm} the convex hull of all (integral) stable matchings. In this section, our main result is $P_{fsm} = P_{sm}$. We say (i, a) are *matched under x* if $x_{ia} > 0$.

Definition 21. Let $x \in P_{fsm}$ and $\Omega_i(x)$ be the set of applicants that are matched to institute i under x . Let $\Omega_i(x)$ be composed of a_{i1}, a_{i2}, \dots , ordered based on the decreasing preference of institute i .

1. Define $H_i(x)$ as a tuple composed of applicants chosen based on the following procedure: adding a_{ij} greedily unless adding the next applicant into $H_i(x)$ will cause $H_i(x)$ to violate the quota of some class. Equivalently, $a_{il} \notin H_i(x)$ only if there exists a class $C_j^i \in a_{il}(\mathbb{C}(i))$ such that $|H_i(x) \cap \{a_{it}\}_{t=1}^{l-1}| = q^+(C_j^i)$.
2. Define $E_i(x)$ as a tuple composed of applicants for whom institute i is the most preferred institute that they are matched under x , i.e., an applicant $a \in E_i(x)$, if $x(T(i, a) \setminus \{(i, a)\}) = 0$ and $x_{ia} > 0$. The order of the applicants in $E_i(x)$ is based on the decreasing preference of institute i .

Lemma 22. $E_i(x)$ is feasible for institute i .

Proof. We need to show that given any class $C_j^i \in \mathbb{C}(i)$, $|E_i(x) \cap C_j^i| \leq q^+(C_j^i)$. We proceed by induction on the height of C_j^i in the tree structure of $\mathbb{C}(i)$. The base case is a leaf class. If $|E_i(x) \cap C_j^i| > q^+(C_j^i)$, form a class-tuple by picking the first $q^+(C_j^i)$ applicants in $E_i(x) \cap C_j^i$. Then Constraint (5) is violated in such a class-tuple. For the induction step, if $|E_i(x) \cap C_j^i| > q^+(C_j^i)$, again choose the $q^+(C_j^i)$ highest-ranking applicants in $E_i(x) \cap C_j^i$ and we claim they form a class-tuple of C_j^i , the reason being that by induction hypothesis, given any $C_k^i \subset C_j^i$, $|E_i(x) \cap C_k^i| \leq q^+(C_k^i)$. Now Constraint (5) is again violated in such a class-tuple. \square

Lemma 23. Suppose that $x \in P_{fsm}$.

- (i) For each institute $i \in \mathcal{I}$, we can find two sets U and V of pairwise disjoint classes so that $U \cup V$ partitions \mathbb{L}^i and all applicants in $\Omega_i(x) \setminus H_i(x)$ belong to the classes in U . Moreover,
 - (ia) $|H_i(x)| = \sum_{C_k^i \in U} q^+(C_k^i) + \sum_{C_k^i \in V} |H_i(x) \cap C_k^i|$;
 - (ib) for each class $C_k^i \in U$, $|H_i(x) \cap C_k^i| = |E_i(x) \cap C_k^i| = q^+(C_k^i)$; for each class $C_k^i \in V$ and each applicant $a \in C_k^i$, if $x_{ia} > 0$, then $x_{ia} = 1$;
 - (ic) for each class $C_k^i \in U$, $\sum_{a \in C_k^i} x_{ia} = q^+(C_k^i)$.
- (ii) For every applicant $a \in H_i(x)$, $x(T(i, a)) = \sum_{i \in \mathcal{I}} x_{ia} = 1$; moreover, given any two institutes $i, i' \in \mathcal{I}$, $H_i(x) \cap H_{i'}(x) = \emptyset$.
- (iii) $|H_i(x)| = |E_i(x)|$ for all institutes $i \in \mathcal{I}$.
- (iv) $\sum_{a \in \mathcal{A}} x_{ia} = |E_i(x)|$ for all institutes $i \in \mathcal{I}$.

Proof. For (i), given any applicant $a \in \Omega_i(x) \setminus H_i(x)$, by Definition 21, there exists some class $C_j^i \in a(\mathbb{C}(i))$ for which $|H_i(x) \cap C_j^i| = q^+(C_j^i)$. Let B be the set of classes C_j^i which contain at least one applicant in $\Omega_i(x) \setminus H_i(x)$ and $|C_j^i \cap H_i(x)| = q^+(C_j^i)$. Let $U := \mathfrak{R}(B)$ and choose V in such a way so that $U \cup V$ partitions \mathbb{L}^i . Now (ia) is a consequence of counting. We will prove (ib)(ic) afterwards.

For (ii), by definition of $H_i(x)$, none of the applicants in $\Omega_i(x) \setminus H_i(x)$ contributes to the shaft $x(S(H_i(x)))$. As a result, for Constraint (3) to hold for the comb $K(i, S(H_i(x)))$, every tooth-applicant $a \in H_i(x)$ must contribute at least 1, and indeed, by Constraint (1), exactly 1. So we have the first

statement of (ii). The second statement holds because it cannot happen that $x(T(i, a)) = x(T(i', a)) = 1$, given that $x_{ia} > 0$ and $x_{i'a} > 0$.

For (iii), By Definition 21, all sets $E_i(x)$ are disjoint; thus, every applicant who is matched under x belongs to exactly one $E_i(x)$ and at most one $H_i(x)$ by (ii). Therefore, $\sum_{i \in \mathcal{I}} |E_i(x)| \geq \sum_{i \in \mathcal{I}} |H_i(x)|$ and we just need to show that for each institute i , $|E_i(x)| \leq |H_i(x)|$, and this follows by using (ia):

$$|H_i(x)| = \sum_{C_k^i \in U} q^+(C_k^i) + \sum_{C_k^i \in V} |H_i(x) \cap C_k^i| \geq \sum_{C_k^i \in U} |E_i(x) \cap C_k^i| + \sum_{C_k^i \in V} |E_i(x) \cap C_k^i| = |E_i(x)|, \quad (6)$$

where the inequality follows from Lemma 22 and the fact all applicants in $\Omega_i(x) \setminus H_i(x)$ are in classes in U . So this establishes (iii). Moreover, as Inequality (6) must hold with equality throughout, for each class $C_k^i \in V$, if applicant $a \in C_k^i$ is matched to institute i under x , he must belong to both $H_i(x)$ and $E_i(x)$, implying $x_{ia} = 1$; given any class $C_k^i \in U$, $|H_i(x) \cap C_k^i| = |E_i(x) \cap C_k^i| = q^+(C_k^i)$. So we have (ib).

For (iv), consider the comb $K(i, S(E_i(x)))$. By definition, $x(T(i, a) \setminus \{(i, a)\}) = 0$ for each applicant $a \in E_i(x)$. So

$$\begin{aligned} x(K(i, S(E_i(x)))) &= x(S(E_i(x))) \\ &= \sum_{C_k^i \in V} |E_i(x) \cap C_k^i| + \sum_{C_k^i \in U} \sum_{a' \in C_k^i, (i, a') \in S(E_i(x))} x_{ia'} \\ &\leq \sum_{C_k^i \in V} |E_i(x) \cap C_k^i| + \sum_{C_k^i \in U} q^+(C_k^i) = |E_i(x)|, \end{aligned}$$

where the inequality follows from Constraint (2) and the rest can be deduced from (ib). By Constraint (3), the above inequality must hold with equality. So for each class $C_k^i \in U$, $\sum_{a' \in C_k^i, (i, a') \in S(E_i(x))} x_{ia'} = \sum_{a' \in C_k^i} x_{ia'} = q^+(C_k^i)$, giving us (ic) and implying that there is no applicant in $C_k^i \in U$ who is matched to institute i under x ranking lower than all applicants in $E_i(x) \cap C_k^i$. The proof of (iv) follows by

$$\sum_{a \in \mathcal{A}} x_{ia} = \sum_{C_k^i \in V} \sum_{a \in C_k^i} x_{ia} + \sum_{C_k^i \in U} \sum_{a \in C_k^i} x_{ia} = \sum_{C_k^i \in V} |E_i(x) \cap C_k^i| + \sum_{C_k^i \in U} q^+(C_k^i) = |E_i(x)|.$$

□

Packing Algorithm

We now introduce a packing algorithm to establish the integrality of the polytope. Our algorithm is generalized from that proposed by Sethuraman, Teo, and Qian [23]. Given $x \in P_{fsm}$, for each institute i , we create $|E_i(x)|$ “bins,” each of size (height) 1; each bin is indexed by (i, j) , where $1 \leq j \leq |E_i(x)|$. Each $x_{ia} > 0$ is an “item” to be packed into the bins. Bins are filled from the bottom to the top. When the context is clear, we often refer to those items x_{ia} as simply applicants; if applicant $a \in C_j^i$, then the item x_{ia} is said to belong to the class C_j^i .

In Phase 0, each institute i puts the items x_{ia} , if $a \in H_i(x)$, into each of its $|E_i(x)|$ bins. In the following phase, $t = 1, 2, \dots$, our algorithm proceeds by

- first finding out the set L_t of bins with maximum available space;

– then assigning each of the bins in L_t one item.

The assignment in each phase proceeds by *steps*, indexed by $l = 1, 2, \dots, |L_t|$. The order of the bins in L_t to be examined does not matter. How the institute i chooses the items to be put into its bins is the crucial part in which our algorithm differs from that of Sethuraman, Teo, and Qian. We maintain the following invariant.

Invariant C : The collection of the least preferred items in the $|E_i(x)|$ bins (e.g., the items currently on top of institute i 's bins) should respect of the quotas of the classes in $\mathbb{C}(i)$.

Subject to this invariant, institute i chooses the best remaining item and adds it into the bin (i, j) , which has the maximum available space in the current phase. This unavoidably raises another issue: how can we be sure that there is at least one remaining item for institute i to put into the bin (i, j) without violating Invariant C ? We will address this issue in our proof.

Theorem 24. *Let $x \in P_{fsm}$. Let $M_{i,j}$ be the set of applicants assigned to bin (i, j) at the end of any step of the packing procedure and $a_{i,j}$ be the lowest-ranking applicant of institute i in bin (i, j) (implying $x_{ia_{i,j}}$ is on top of bin (i, j)). Then*

- (i) *In any step, suppose that the algorithm is examining bin (i, j) . Then institute i can find at least one item in its remaining items to add into bin (i, j) without violating Invariant C ;*
- (ii) *For all bins (i, j) , $x(M_{i,j} \setminus \{a_{i,j}\}) + x(T(i, a_{i,j}) \setminus \{(i, a_{i,j})\}) = 1$;*
- (iii) *At the end of any step, institute i can organize a comb $K(i, S(A^i))$ where A^i is composed of applicants in $\{a_{i,j'}\}_{j'=1}^{|E_i(x)|}$ so that $x(K(i, S(A^i))) = \sum_{j'=1}^{|E_i(x)|} x(M_{i,j'}) + \sum_{j'=1}^{|E_i(x)|} x(T(i, a_{i,j'}) \setminus \{(i, a_{i,j'})\}) = |E_i(x)|$;*
- (iv) *At the end of any step, an item x_{ia} is not put into institute i 's bins if and only if there exists a class $C_{at}^i \in a(\mathbb{C}(i))$ so that $|\{a_{i,j'}\}_{j'=1}^{|E_i(x)|} \cap C_{at}^i \cap \mathbb{L}_{\succ_a}^i| = q^+(C_{at}^i)$.*
- (v) *If x_{ia} is packed and $x_{i'a}$ is not, then $i' \succ_a i$;*
- (vi) *At the end of any phase, the $a_{i,j}$ in all bins are distinct. In particular, for any applicant a who is matched under x , there exists some bin (i, j) such that $a = a_{i,j}$.*

Proof. We first assume that (ii) holds and prove (i). Observe that (ii) implies that given any applicant $a \in E_i(x)$, its corresponding item x_{ia} , if already put into a bin, must be on its top and fills it completely. Since (i, j) currently has available space, at least one applicant in $E_i(x)$ is not in institute i 's bins yet. We claim that there exists at least one remaining applicant in $E_i(x)$ that can be added into bin (i, j) . Suppose not. Let the set of applicants in $E_i(x)$ that are not put into i 's bins be G . Given any applicant $a \in G$, there must exist some class $C_k^i \in a(\mathbb{C}(i))$ for which $|\bigcup_{1 \leq j' \leq |E_i(x)|, j' \neq j} a_{i,j'} \cap C_k^i| = q^+(C_k^i)$. Let B be the set of classes C_k^i that contains at least one applicant in G and $|\bigcup_{1 \leq j' \leq |E_i(x)|, j' \neq j} a_{i,j'} \cap C_k^i| = q^+(C_k^i)$. Let G' be $(E_i(x) \setminus G) \setminus \bigcup_{C_k^i \in \mathfrak{R}(B)} C_k^i$, the subset of applicants in $E_i(x)$ that are already put into the bins but not belonging to any class in $\mathfrak{R}(B)$. Note that none of the applicants in G' can be in the bin (i, j) . Thus, by counting the number of the bins minus (i, j) , we have

$$|E_i(x)| - 1 \geq |G'| + \sum_{C_k^i \in \mathfrak{R}(B)} \left| \bigcup_{j'=1, j' \neq j}^{|E_i(x)|} a_{i,j'} \cap C_k^i \right| = |G'| + \sum_{C_k^i \in \mathfrak{R}(B)} q^+(C_k^i)$$

Note that all applicants in $E_i(x) \setminus G'$ are in some class in $\mathfrak{R}(B)$ (either they are already put into the bins or not). Then by the pigeonhole principle, there is at least one class $C_k^i \in \mathfrak{R}(B)$ for which $|(E_i(x) \setminus G') \cap C_k^i| > q^+(C_k^i)$, contradicting Lemma 22.

We now prove (ii)-(vi) by induction on the number of phases. In the beginning, (ii)(v)(vi) holds by Lemma 23(ii)(iii). (iii)(iv) hold by setting $A^i := H_i(x)$ and observation Definition 21 and Lemma 23(ii). Suppose that the theorem holds up to Phase t . Let α be the maximum available space in Phase $t + 1$. Suppose that the algorithm is examining bin (i, j) and institute i chooses item x_{ia} to be put into this bin. From (vi) of the induction hypothesis, applicant a is on top of another bin (i', j') , where $i' \neq i$, in the beginning of phase $t + 1$. Then by (ii)(v) of the induction hypothesis,

$$x(T(i, a)) \leq x(T(i', a)) - x_{i'a} = 1 - x(M_{i', j'}) \leq \alpha, \quad (7)$$

where the last inequality follows from our assumption that in Phase $t + 1$, the maximum available space is α . Note also that

$$x(T(i, a)) = \alpha, \text{ then } (i', j') \in L_{t+1} \text{ (bin } (i', j') \text{ is also examined in Phase } t + 1). \quad (8)$$

Assume that \bar{A}^i is a tuple composed of applicants in $\{a_{i, j'}\}_{j'=1}^{|E_i(x)|}$. For our induction step, let $A^i := \bar{A}^i |^{a_{i, j} a}$, the new set of items on top of i 's bins after a is put on top of $a_{i, j}$.

We first prove (iv). Since x_{ia} is not put into the bin before this step, by (iv) of the induction hypothesis, there exists some class $C_{al}^i \in a(\mathbb{C}(i))$ for which $|\bar{A}^i \cap C_{al}^i \cap \mathbb{L}_{>a}^i| = q^+(C_{al}^i)$. Let C_{al}^i be the smallest such class. Since x_{ia} is allowed to put on top of $x_{ia_{i, j}}$, $a_{ij} \succ_i a$ and $a_{ij} \in C_{al}^i$, otherwise, Invariant C regarding $q^+(C_{al}^i)$ is violated.

Now we show that all other items $x_{ia'}$ fulfill the condition stated in (iv). There are two cases.

- Suppose that $x_{ia'}$ is not put into the bins yet.
 - Suppose that $a_{i, j} \succ_i a' \succ_i a$. We claim that it cannot happen that for all classes $C_{a't}^i \in a'(\mathbb{C}(i))$, $|A^i \cap C_{a't}^i \cap \mathbb{L}_{>a'}^i| < q^+(C_{a't}^i)$, otherwise, $A^i |^{a'}$ is still feasible, in which case institute i would have chosen $x_{ia'}$, instead of x_{ia} to put into bin (i, j) , a contradiction.
 - Suppose that $a_{i, j} \succ_i a \succ_i a'$. By (iv) of the induction hypothesis, there exists a class $C_{a't'}^i \in a'(\mathbb{C}(i))$ for which $|\bar{A}^i \cap C_{a't'}^i \cap \mathbb{L}_{>a'}^i| = q^+(C_{a't'}^i)$. If $C_{a't'}^i \not\subset C_{al}^i$, it is easy to see that $|A^i \cap C_{a't'}^i \cap \mathbb{L}_{>a'}^i| = q^+(C_{a't'}^i)$; if $C_{a't'}^i \subset C_{al}^i$, then $C_{al}^i \in a'(\mathbb{C}(i))$ and we have $|A^i \cap C_{al}^i \cap \mathbb{L}_{>a'}^i| = q^+(C_{al}^i)$. In both situations, the condition of (iv) regarding $x_{ia'}$ is satisfied.
- Suppose that $x_{ia'}$ is already put into the bins. It is trivial if $a' \succ_i a$, so assume that $a \succ_i a'$. We claim that none of the classes $C_{a't}^i \in a'(\mathbb{C}(i))$ can be a subclass of C_{al}^i or C_{al}^i itself. Otherwise, $C_{al}^i \in a'(\mathbb{C}(i))$, and we have $q^+(C_{al}^i) = |\bar{A}^i \cap C_{al}^i \cap \mathbb{L}_{>a}^i| \geq |\bar{A}^i \cap C_{al}^i \cap \mathbb{L}_{>a'}^i|$, a contradiction to (iv) of the induction hypothesis. Now since for every class $C_{a't}^i \in a'(\mathbb{C}(i))$, we have $C_{a't}^i \not\subset C_{al}^i$, we have $|A^i \cap C_{a't}^i \cap \mathbb{L}_{>a'}^i| = |\bar{A}^i \cap C_{a't}^i \cap \mathbb{L}_{>a'}^i| < q^+(C_{a't}^i)$, where the strict inequality is due to the induction hypothesis.

We notice that the quantity $\sum_{j'=1}^{|E_i(x)|} x(M_{i, j'})$ is exactly the sum of the shaft $x(S(\bar{A}^i))$ (before x_{ia} is added) or $x(S(A^i))$ (after x_{ia} is added) by observing (iv). Below let $x(\bar{M}_{i, j})$ and $x(M_{i, j})$ denote the total size of the items in bin (i, j) before and after x_{ia} is added into it. So $x(M_{i, j}) = x(\bar{M}_{i, j}) + x_{ia}$. Now we can derive the following:

$$\begin{aligned}
x(K(i, S(A^i))) &= x(S(A^i)) + x(T(i, a) \setminus \{(i, a)\}) + \sum_{j'=1, j' \neq j}^{|E_i(x)|} x(T(i, a_{i,j'}) \setminus \{(i, a_{i,j'})\}) \\
&= x(\overline{M}_{i,j}) + x_{ia} + x(T(i, a) \setminus \{(i, a)\}) + \sum_{j'=1, j' \neq j}^{|E_i(x)|} x(M_{i,j'}) + x(T(i, a_{i,j'}) \setminus \{(i, a_{i,j'})\}) \\
&= x(\overline{M}_{i,j}) + x(T(i, a)) + |E_i(x)| - 1 \quad (\text{by (ii) of the induction hypothesis}) \\
&\geq |E_i(x)| \quad (\text{by Constraint (3)})
\end{aligned}$$

For the above inequality to hold,

$$x(\overline{M}_{i,j}) + x(T(i, a)) \geq 1. \quad (9)$$

Since $x(\overline{M}_{i,j}) = 1 - \alpha$ and $x(T(i, a)) \leq \alpha$ by Inequality (7), Inequality (9) must hold with equality, implying that $x(K(i, S(A^i))) = |E_i(x)|$, giving us (iii).

Since institute i puts x_{ia} into bin (i, j) , the “new” $M_{i,j}$ and the “new” $a_{i,j}$ ($=a$) satisfies

$$x(M_{i,j}) + x(T(i, a) \setminus \{(i, a)\}) = 1.$$

This establishes (ii). (v) follows because Inequality (7) must hold with equality throughout. Therefore, there is no institute i'' which ranks strictly between i and i' and $x_{i''a} > 0$.

Finally for (vi), note that $x(T(i, a)) = \alpha$ if the item x_{ia} is put into some bin in Phase $t+1$. All such items are the least preferred items in their respective “old” bins (immediately before Phase $t+1$), it means the items on top of the newly-packed bins are still distinct. Moreover, from (8), if a bin (i, j) is not examined in Phase $t+1$, then its least preferred applicant cannot be packed in phase $t+1$ either. \square

We define an assignment μ^α based on a number $\alpha \in [0, 1)$ as follows. Assume that there is a line of height α “cutting through” all the bins horizontally. If an item x_{ia} whose position in i ’s bins intersects α , applicant a is assigned to institute i . In the case this cutting line of height α intersects two items in the same bin, we choose the item occupying the higher position. More precisely:

Given $\alpha \in [0, 1)$, for each institute $i \in \mathcal{I}$, we define an assignment as follows: $\mu^\alpha(i) = \{a : 1 - x(T(i, a)) \leq \alpha < 1 - x(T(i, a)) + x_{ia}\}$.

Theorem 25. *The polytope determined by Constraints (1)-(5) is integral.*

Proof. We generate uniformly at random a number $\alpha \in [0, 1)$ and use it to define an assignment μ^α . To facilitate the discussion, we choose the largest $\alpha' \leq \alpha$ so that $\mu^{\alpha'} = \mu^\alpha$. Intuitively, this can be regarded as lowering the cutting line from α to α' without modifying the assignment, and $1 - \alpha'$ is exactly the maximum available space in the beginning of a certain phase l during the execution of our packing algorithm. Note that the assignment μ^α is then equivalent to giving those applicants (items) on top of institute i ’s bins to i at the end of phase l .

We now argue that μ^α is a stable matching. First, it is a matching by Theorem 24(vi). The matching respects the quota of all classes since Invariant C is maintained. What remains to be argued is the stability of μ^α . Suppose, for a contradiction, (i, a^ϕ) is a blocking pair. We consider the possible cases.

- Suppose that $x_{ia^\phi} > 0$ and x_{ia^ϕ} is not put into the bins yet at the end of Phase l . Then by Theorem 24(iv) and the definition of blocking pairs, (i, a^ϕ) cannot block μ^α .
- Suppose that $x_{ia^\phi} > 0$ and x_{ia^ϕ} is already put into the bins at the end of Phase l . If $\mu^\alpha(a^\phi) = i$, there is nothing to prove. So assume $\mu^\alpha(a^\phi) \neq i$ and this means that the item x_{ia^ϕ} is “buried” under some other item on top of some of i ’s bins at the end of Phase l . Then by Theorem 24(v), a^ϕ is assigned to some other institute ranking higher than i , contradicting the assumption that (i, a^ϕ) is a blocking pair.
- Suppose that $x_{ia^\phi} = 0$. There are two subcases.
 - Suppose that for each of the classes $C_{a^\phi t}^i \in a^\phi(\mathbb{C}(i))$, $|\mu^\alpha(i) \cap C_{a^\phi t}^i| < q^+(C_{a^\phi t}^i)$. Then we can form a new feasible tuple $\mu^\alpha(i)|a^\phi$. It can be inferred from the definition of the shaft that $x(S(\mu^\alpha(i)|a^\phi)) \leq x(S(\mu^\alpha(i)))$. Moreover, by Theorem 24(iii), we have $x(K(i, S(\mu^\alpha(i)))) = |E_i(x)|$. Now by Constraint (3),

$$\begin{aligned}
|E_i(x)| + 1 &\leq x(K(i, S(\mu^\alpha(i)|a^\phi))) \\
&\leq x(S(\mu^\alpha(i))) + x(T(i, a^\phi) \setminus \{(i, a^\phi)\}) + \sum_{a \in \mu^\alpha} x(T(i, a) \setminus \{(i, a)\}) \\
&= x(K(i, S(\mu^\alpha(i)))) + x(T(i, a^\phi) \setminus \{(i, a^\phi)\}) \\
&= |E_i(x)| + x(T(i, a^\phi) \setminus \{(i, a^\phi)\}).
\end{aligned}$$

As a result, $x(T(i, a^\phi) \setminus \{(i, a^\phi)\}) = 1$, implying that $\mu^\alpha(a^\phi) \succ_{a^\phi} i$, a contradiction to the assumption that (i, a) blocks μ^α .

- Suppose that there exists a class $C_{a^\phi l^\phi}^i \in a^\phi(\mathbb{C}(i))$ for which $|\mu^\alpha(i) \cap C_{a^\phi l^\phi}^i| = q^+(C_{a^\phi l^\phi}^i)$. Let $C_{a^\phi l^\phi}^i$ be the smallest such class. By definition of blocking pairs, there must exist an applicant $a^\dagger \in \mu^\alpha(i) \cap C_{a^\phi l^\phi}^i$ who ranks lower than a^ϕ . Choose a^\dagger to be the lowest ranking such applicant in $\mu^\alpha(i)$. We make the following critical observation:

$$x(S(\mu^\alpha(i)|a^\dagger a^\phi)) \leq x(S(\mu^\alpha(i))) - x_{ia^\dagger}. \quad (10)$$

To see this, we first argue that given an item $x_{ia} > 0$, if it does not contribute to the shaft $S(\mu^\phi(i))$, then it cannot contribute to shaft $S(\mu^\alpha(i)|a^\dagger a^\phi)$ either. It is trivial if $a \succ_i a^\dagger$. So assume that $a^\dagger \succ_i a$. First suppose that $a \notin C_{a^\phi l^\phi}^i$. Then given any class $C_{at}^i \in a(\mathbb{C}(i))$, $|\mu^\alpha(i) \cap C_{at}^i \cap \mathbb{L}_{\succ_a}^i| = |\mu^\alpha(i)|^{a^\dagger} a^\phi \cap C_{at}^i \cap \mathbb{L}_{\succ_a}^i|$, and Theorem 24(iv) states that there is a class $C_{al}^i \in a(\mathbb{C}(i))$ such that $|\mu^\alpha(i) \cap C_{al}^i \cap \mathbb{L}_{\succ_a}^i| = q^+(C_{al}^i)$. Secondly suppose that $a \in C_{a^\phi l^\phi}^i$. Observe that $q^+(C_{a^\phi l^\phi}^i) = |\mu^\alpha(i)|^{a^\dagger} a^\phi \cap C_{a^\phi l^\phi}^i \cap \mathbb{L}_{\succ_{a^\dagger}}^i| = |\mu^\phi(i)|^{a^\dagger} a^\phi \cap C_{a^\phi l^\phi}^i \cap \mathbb{L}_{\succ_a}^i|$ (the first equality follows from the choice of a^\dagger). In both cases, we conclude that x_{ia} cannot contribute to the shaft $S(\mu^\phi(i)|a^\dagger a^\phi)$. The term x_{ia^\dagger} does not contribute to the shaft $S(\mu^\phi(i)|a^\dagger a^\phi)$ by the same argument. Now using Constraint (3), Theorem 24(iii), and Inequality (10), we have

$$\begin{aligned}
|E_i(x)| &\leq x(K(i, S(\mu^\alpha(i)|a^\dagger a^\phi))) \\
&\leq x(S(\mu^\alpha(i))) - x_{ia^\dagger} + x(T(i, a^\phi) \setminus \{(i, a^\phi)\}) + \sum_{a \in \mu^\alpha(i) \setminus \{a^\dagger\}} x(T(i, a) \setminus \{(i, a)\}) \\
&= |E_i(x)| - x(T(i, a^\dagger)) + x(T(i, a^\phi)). \quad (\text{Note that } x_{ia^\phi} = 0).
\end{aligned}$$

Therefore,

$$x(T(i, a^\phi)) \geq x(T(i, a^\dagger)) \geq 1 - \alpha' \geq 1 - \alpha.$$

So $\mu^\alpha(a^\phi) \succ_{a^\phi} i$, again a contradiction to the assumption that (i, a^ϕ) blocks μ^α .

So we have established that the generated assignment μ^α is a stable matching. Now the remaining proof is the same as in [24]. Assume that $\mu^\alpha(i, a) = 1$ if and only if applicant a is assigned to institute i under μ^α . Then

$$\text{Exp}[\mu^\alpha(i, a)] = x_{ia}.$$

Then $x_{ia} = \int_0^1 \mu^\alpha(i, a) d\alpha$ and x can be written as a convex combination of μ^α as α varies over the interval $[0, 1)$. The integrality of the polytope thus follows. \square

4.1 Optimal Stable Matching

Since our polytope is integral, we can write suitable objective functions to target for various optimal stable matchings using Ellipsoid algorithm [11]. As the proposed LP has an exponential number of constraints, we also design a separation oracle to get a polynomial time algorithm. The basic idea of our oracle is based on dynamic programming.

4.2 Median-Choice Stable Matching

An application of our polyhedral result is the following.

Theorem 26. *Suppose that in the given instance, all classifications are laminar families and there is no lower bound, $q^-(C_j^i) = 0$ for any class C_j^i . Let $\mu_1, \mu_2, \dots, \mu_k$ be stable matchings. If we assign every applicant to his median choice among all the k matchings, the outcome is a stable matching.*

Proof. Let x_{μ_t} be the solution based on μ_t for any $1 \leq t \leq k$ and apply our packing algorithm on the fractional solution $x = \frac{\sum_{t=1}^k x_{\mu_t}}{k}$. Then let $\alpha = 0.5$ and $\mu^{0.5}$ be the stable matching resulted from the cutting line of height $\alpha = 0.5$. We make the following observation based on Theorem 24:

Suppose that applicant a is matched under x and those institutes with which he is matched are $i_1, i_2, \dots, i_{k'}$, ordered based on their rankings on a 's preference list. Assume that he is matched to i_t n_t times among the k given stable matchings. At the termination of the packing algorithm, each of the items $x_{i_l a}$, $1 \leq l \leq k'$, appears in institute i_l 's bins and its position is from $\sum_{t=1}^{l-1} \frac{n_t}{k}$ to $\sum_{t=1}^l \frac{n_t}{k}$.

Now $\mu^{0.5}$ gives every applicant his median choice follows easily from the above observation. \square

Using similar ideas, we can show that an applicant-optimal stable matching must be institute-(lexicographical)-pessimal and similarly an applicant-pessimal stable matching must be institute-(lexicographical)-optimal: by taking x as the average of all stable matchings and consider the two matching μ^ϵ and $\mu^{1-\epsilon}$ with arbitrary small $\epsilon > 0$. Hence, it is tempting to conjecture that the median choice stable matching is also a lexicographical median outcome for the institutes. Somehow surprisingly, it turns out not to be the case and a counter-example can be found in the appendix.

4.3 Polytope for Many-to-Many “Unclassified” Stable Matching

In the many-to-many stable matching problem, each entity $e \in \mathcal{I} \cup \mathcal{A}$ has a quota $Q(e) \in \mathbb{Z}^+$ and a preference over a subset of the other side. A matching μ is feasible if given any entity $e \in \mathcal{I} \cup \mathcal{A}$, (1) $|\mu(e)| \leq Q(e)$, and (2) $\mu(e)$ is a subset of the entities on e 's preference list. A feasible matching μ is stable if there is no blocking pair (i, a) , which means that i prefers a to one of the assignments $\mu(i)$, or if $|\mu(i)| < Q(i)$ and $a \notin \mu(i)$; and similarly a prefers i to one of his assignments $\mu(a)$, or if $|\mu(a)| < Q(a)$ and $i \notin \mu(a)$.

We now transform the problem into (many-to-one) **LCSM**. For each applicant $a \in \mathcal{A}$, we create $Q(a)$ copies, each of which retains the original preference of a . All institutes replace the applicants by their clones on their lists. To break ties, all institutes rank the clones of the same applicant in an arbitrary but fixed manner. Finally, each institute treats the clones of the same applicant as a class with upper bound 1. It can be shown that the stable matchings in the original instance and in the transformed **LCSM** instance have a one-one correspondence. Thus, we can use Constraints (1)-(5) to describe the former⁶.

5 Conclusion and Future Work

In this paper, we introduce **CLASSIFIED STABLE MATCHING** and present a dichotomy theorem to draw a line between its polynomial solvability and NP-completeness. We also study the problem using the polyhedral approach and propose polynomial time algorithms to obtain various optimal matchings.

We choose the terms “institutes” and “applicants” in our problem definition, instead of the more conventional hospitals and residents, for a reason. We are aware that in real-world academics, many departments not only have ranking over their job candidates but also classify them based on their research areas. When they make their hiring decision, they have to take the quota of the classes into consideration. And in fact, we were originally motivated by this common practice.

CLASSIFIED STABLE MATCHING has happened in real world. In a hospitals/residents matching program in Scotland, certain hospitals declared that they did not want more than one female physician⁷. Roth [17] proposed an algorithm to show that stable matchings always exist.

There are quite a few questions that remain open. The obvious one would be to write an LP to describe **LCSM** with both upper bounds and lower bounds. Even though we can obtain various optimal stable matchings, the Ellipsoid algorithm can be inefficient. It would be nicer to have fast combinatorial algorithms. The *rotation* structure of Gusfield and Irving [12] seems the way to go.

Acknowledgments

I thank Peter Winkler for suggesting to me the idea of **CLASSIFIED STABLE MATCHING** and Alvin Roth and Jay Sethuraman for several fruitful discussions. The idea of reducing many-to-many stable matching to **LCSM** was suggested by Jay Sethuraman. The comments of SODA 10 reviewers have been extremely helpful.

⁶ This cloning technique works for the many-to-one **HOSPITALS/RESIDENTS** problem, i.e, we can clone the hospitals and use the original one-one stable marriage polytope [25] to describe that of the former [21, 23]. However, this trick does not work here for many-to-many matching. The reason is that the same applicant cannot be assigned to an institute multiple times. And that is why our classifications help to resolve this problem.

⁷ However the restriction has not been in place since at least 1997, and possibly longer [1].

References

1. Private Communication.
2. ABELEDO, H. G., AND ROTHBLUM, U. G. Stable matchings and linear inequalities. *Discrete Applied Mathematics* 54(1) (1994), 1–27.
3. ABRAHAM, D., IRVING, R., AND MANLOVE, D. Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms* 5(1) (2007), 73–90.
4. BAÏOU, M., AND BALINSKI, M. The stable admissions polytope. *Mathematical Programming, series A* 87 (2000), 427–439.
5. BIRÓ, P., FLEINER, T., IRVING, R., AND MANLOVE, D. The college admissions problem with lower and common quotas. University of Glasgow, Computing Science Department Research Report, TR-2009-303, 2009.
6. FLEINER, T. Some results on stable matchings and fixed points, 2002. TR-2002-08, EGRES, December.
7. FLEINER, T. A fixed-point approach to stable matchings and some applicants. *Mathematical Operations Research* 28(1) (2003), 103–126.
8. GALE, D., AND SHAPLEY, L. College admissions and the stability of marriage. *American Mathematical Monthly* 69(1) (1962), 9–15.
9. GALE, D., AND SOTOMAYOR, M. Some remarks on the stable matching problem. *Discrete Applied Mathematics* 11 (1985), 223–232.
10. GAREY, M., AND JOHNSON, D. *Computers and Intractability*. Freeman, 1979.
11. GRÖTSCHEL, M., LOVÁSZ, L., AND SCHRIJVER, A. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1(2) (1981), 70–89.
12. GUSFIELD, D., AND IRVING, R. *The Stable Marriage Problem*. The MIT Press, 1989.
13. HAMADA, K., MIYAZAKI, S., AND IWAMA, K. The hospitals/residents problem with quota lower bounds. In *Matching Up: Matching Under Preferences-Algorithms and Complexity, Satellite workshop of ICALP 2008* (2008), pp. 55–66.
14. KLAUS, B., AND KLIJN, F. Median stable matching for college admissions. *International Journal of Game Theory* 34(1) (2006), 1–11.
15. KNUTH, D. *Mariages stables et leurs relations avec d’autres problèmes combinatoires*. Les Presses de l’université de Montréal, 1976.
16. ROTH, A. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica* 54 (1986), 425–427.
17. ROTH, A. A natural experiment in the organization of entry level labor markets: Regional markets for new physicians and surgeons in the U.K. *American Economic Review* 81 (1991), 415–440.
18. ROTH, A., ROTHBLUM, U. G., AND VATE, J. H. V. Stable matchings, optimal assignments and linear programming. *Mathematics of operation research* 18 (1993), 808–828.
19. ROTH, A., AND SOTOMAYOR, M. *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1990.
20. ROTHBLUM, U. G. Characterization of stable matchings as extreme points of a polytope. *Mathematical Programming* 54 (1992), 1475–1480.
21. SETHURAMAN, J. Private communication, 2009.
22. SETHURAMAN, J., AND TEO, C.-P. A polynomial-time algorithm for the bistable roommates problem. *Journal of Computer and System Sciences* 63(3) (2001), 486–497.
23. SETHURAMAN, J., TEO, C.-P., AND QIAN, L. Many-to-one stable matching: Geometry and fairness. *Mathematics of Operations Research* 31(3) (2006), 581–596.
24. TEO, C.-P., AND SETHURAMAN, J. The geometry of fractional stable matchings and its applications. *Mathematics of Operations Research* 23(4) (2001), 874–891.
25. VATE, J. H. V. Linear programming brings marital bliss. *Operation Research Letters* 8 (1989), 147–153.

A An Example for Section 2.2

In contrast to the generalized rural hospitals theorem in **LCSM**, if some institutes use intersecting classes, stable matching sizes may differ. Figure 3 is an example.

Institute Preferences	Classifications	Quota
$i_1: a_1 a_2 a_3$	$C_1^1 = \{a_1, a_2\}, C_2^1 = \{a_1, a_3\}$	$Q(i_1) = 2, q^+(C_1^1) = 1, q^+(C_2^1) = 1$
$i_2: a_2 a_1 a_3 a_4$	$C_1^2 = \{a_2, a_1\}, C_2^2 = \{a_2, a_3\}, C_3^2 = \{a_2, a_4\}$	$Q(i_2) = 2, q^+(C_1^2) = 1, q^+(C_2^2) = 1, q^+(C_3^2) = 1$
Applicant Preferences		
$a_1: i_1 i_2$		
$a_2: i_1 i_2$		
$a_3: i_1 i_2$		
$a_4: i_2$		
Stable Matchings		
$\mu_A = \{(i_1; a_1), (i_2; a_2)\}$		
$\mu_B = \{(i_1; a_2, a_3), (i_2; a_1, a_4)\}$		

Fig. 3. An example of stable matchings of different sizes.

B Missing Proofs of Section 3

In this section, we prove Theorem 16. We assume that the set of posets $\mathbb{P} = \{P_1, P_2, \dots, P_k\}$ contains a poset which is not a downward forest. Moreover, we assume that there is no lower bound on the classes.

Without loss of generality, we assume that P_1 is not a downward forest. Such a poset must have a “V.” By definition, there exists institute i whose class inclusion poset $P(i)$ is isomorphic to P_1 . This implies that institute i must have two intersecting classes in $\mathbb{C}(i)$. In the following, we will present a reduction in which all institutes use *at most* two classes (that can be intersecting). It is straightforward to use some dummy institutes and applicants to “pad” our reduction so that every poset $P_j \in \mathbb{P}$ is isomorphic to some class inclusion poset of the institutes in the derived instance. Our reduction is from ONE-IN-THREE-SAT. We will use an instance in which there is no negative literal. (NP-completeness still holds under this restriction [10].)

The overall goal is to design a reduction so that the derived \mathbb{P} -CLASSIFIED STABLE MATCHING instance allows a stable matching if and only if the given instance $\phi = c_1 \wedge c_2 \wedge \dots \wedge c_k$ is satisfiable. We will build a set of *clause gadgets* to represent each clause c_j . For every pair of literals which belong to the same clause, we create a *literal-pair* gadget. Such a gadget will guarantee that at most one literal it represents can be “activated” (set to TRUE). The clause gadget interacts with the literal-pair gadgets in such a way that if the clause is to be satisfied, exactly one literal it contains can be activated.

Literal-Pair Gadget Suppose that x_i^j and $x_{i'}^j$ both belong to the same clause c_j . We create a gadget $\Upsilon_{i,i'}^j$, composed of four applicants $\{a_{i,t}^j\}_{t=1}^2 \cup \{a_{i',t}^j\}_{t=1}^2$ and two institutes $\{I_i^j, I_{i'}^j\}$ whose preferences and classifications are summarized below.

$$\begin{array}{l|l}
a_{i,1}^j: I_i^j \succ \Gamma(a_{i,1}^j) \succ I_{i'}^j & I_i^j: a_{i,2}^j \succ a_{i,1}^j \succ a_{i',2}^j \succ a_{i',1}^j \succ \Psi(I_i^j) \quad C_1^{I_i^j} = \{a_{i,1}^j, a_{i,2}^j\}, C_2^{I_i^j} = \{a_{i,1}^j, a_{i',1}^j\} \\
a_{i,2}^j: I_{i'}^j \succ I_i^j & Q(I_i^j) = 2, q^+(C_1^{I_i^j}) = 1, q^+(C_2^{I_i^j}) = 1 \\
a_{i',1}^j: I_i^j \succ \Gamma(a_{i',1}^j) \succ I_{i'}^j & I_{i'}^j: a_{i,1}^j \succ a_{i,2}^j \succ a_{i',1}^j \succ a_{i',2}^j \quad C_1^{I_{i'}^j} = \{a_{i,1}^j, a_{i,2}^j\} \\
a_{i',2}^j: I_{i'}^j \succ I_i^j & Q(I_{i'}^j) = 2, q^+(C_1^{I_{i'}^j}) = 1
\end{array}$$

We postpone the explanation of the Γ and Ψ functions for the time being. We first make the following claim.

Claim B: Suppose that in a stable matching μ , the only possible assignments for $\{a_{i,1}^j, a_{i,2}^j, a_{i',1}^j, a_{i',2}^j\}$ are $\{I_i^j, I_{i'}^j\}$. Then there can only be three possible outcomes in μ .

1. $\mu(a_{i,1}^j) = I_i^j, \mu(a_{i,2}^j) = I_{i'}^j, \mu(a_{i',1}^j) = I_{i'}^j, \mu(a_{i',2}^j) = I_i^j$. (In this case, we say x_i is activated while $x_{i'}$ remains deactivated.)
2. $\mu(a_{i,1}^j) = I_{i'}^j, \mu(a_{i,2}^j) = I_i^j, \mu(a_{i',1}^j) = I_i^j, \mu(a_{i',2}^j) = I_{i'}^j$. (In this case, we say $x_{i'}$ is activated while x_i remains deactivated.)
3. $\mu(a_{i,1}^j) = I_{i'}^j, \mu(a_{i,2}^j) = I_i^j, \mu(a_{i',1}^j) = I_{i'}^j, \mu(a_{i',2}^j) = I_i^j$. (In this case, we say both x_i and $x_{i'}$ remain deactivated.)

Claim B can be easily verified. Note that the case $\mu(a_{i,1}^j) = I_i^j, \mu(a_{i,2}^j) = I_{i'}^j, \mu(a_{i',1}^j) = I_i^j, \mu(a_{i',2}^j) = I_{i'}^j$ will not happen due to the quota $q^+(C_2^{I_i^j})$. This case corresponds to the situation that x_i and $x_{i'}$ are both activated and is what we want to avoid.

We now explain how to realize the supposition in Claim B about the fixed potential assignments for $\{a_{i,t}^j\}_{t=1}^2 \cup \{a_{i',t}^j\}_{t=1}^2$ in a stable matching. It can be easily checked that if $a_{i,1}^j$ is matched to some institute in $\Gamma(a_{i,1}^j)$, or either of $\{a_{i,1}^j, a_{i,2}^j\}$ is unmatched; or if either of $\{a_{i',1}^j, a_{i',2}^j\}$ is unmatched, then there must exist a blocking group involving a subset of $\{I_i^j, I_{i'}^j, \{a_{i,t}^j\}_{t=1}^2, \{a_{i',t}^j\}_{t=1}^2\}$. However, the following matching μ^ϕ can happen in which $a_{i',1}^j$ is matched to some institute in $\Gamma(a_{i',1}^j)$ but there is no blocking group: $\mu^\phi(a_{i,1}^j) = I_i^j, \mu^\phi(a_{i,2}^j) = \mu^\phi(a_{i',2}^j) = I_{i'}^j, \mu^\phi(a_{i',1}^j) \in \Gamma(a_{i',1}^j)$.⁸

To prevent the above scenario from happening (i.e., we want μ^ϕ to be unstable), we introduce another gadget \bar{T}_i^j , associated with I_i^j , to guarantee a blocking group will appear. We now list the preferences and classifications of the members of \bar{T}_i^j below.

$$\begin{array}{l|l}
\bar{a}_{i,1}^j: \bar{T}_{i,4}^j \succ \bar{T}_{i,1}^j \succ \bar{T}_{i,3}^j \succ \bar{T}_{i,2}^j & \bar{T}_{i,1}^j: \bar{a}_{i,5}^j \succ \bar{a}_{i,2}^j \succ \bar{a}_{i,4}^j \succ \bar{a}_{i,6}^j \succ \bar{a}_{i,3}^j \succ \bar{a}_{i,1}^j \quad Q(\bar{T}_{i,1}^j) = 2 \\
\bar{a}_{i,2}^j: \bar{T}_{i,3}^j \succ \bar{T}_{i,4}^j \succ \bar{T}_{i,2}^j \succ \bar{T}_{i,1}^j & \bar{T}_{i,2}^j: \bar{a}_{i,4}^j \succ \bar{a}_{i,6}^j \succ \bar{a}_{i,2}^j \succ \bar{a}_{i,3}^j \succ \bar{a}_{i,1}^j \succ \bar{a}_{i,5}^j \quad C_1^{\bar{T}_{i,2}^j} = \{\bar{a}_{i,1}^j, \bar{a}_{i,2}^j, \bar{a}_{i,3}^j\}, C_2^{\bar{T}_{i,2}^j} = \{\bar{a}_{i,3}^j, \bar{a}_{i,4}^j, \bar{a}_{i,5}^j\} \\
\bar{a}_{i,3}^j: \bar{T}_{i,4}^j \succ \bar{T}_{i,3}^j \succ \bar{T}_{i,1}^j \succ \bar{T}_{i,2}^j & Q(\bar{T}_{i,2}^j) = 2, q^+(C_1^{\bar{T}_{i,2}^j}) = 1, q^+(C_2^{\bar{T}_{i,2}^j}) = 1 \\
\bar{a}_{i,4}^j: \bar{T}_{i,4}^j \succ \bar{T}_{i,1}^j \succ \bar{T}_{i,2}^j \succ \bar{T}_{i,3}^j & \bar{T}_{i,3}^j: \bar{a}_{i,4}^j \succ \bar{a}_{i,5}^j \succ \bar{a}_{i,6}^j \succ \bar{a}_{i,3}^j \succ \bar{a}_{i,1}^j \succ \bar{a}_{i,2}^j \quad C_1^{\bar{T}_{i,3}^j} = \{\bar{a}_{i,1}^j, \bar{a}_{i,2}^j, \bar{a}_{i,3}^j\}, C_2^{\bar{T}_{i,3}^j} = \{\bar{a}_{i,3}^j, \bar{a}_{i,4}^j, \bar{a}_{i,5}^j\} \\
\bar{a}_{i,5}^j: \bar{T}_{i,2}^j \succ \bar{T}_{i,4}^j \succ \bar{T}_{i,3}^j \succ \bar{T}_{i,1}^j & Q(\bar{T}_{i,3}^j) = 2, q^+(C_1^{\bar{T}_{i,3}^j}) = 1, q^+(C_2^{\bar{T}_{i,3}^j}) = 1 \\
\bar{a}_{i,6}^j: \bar{T}_{i,2}^j \succ \bar{T}_{i,4}^j \succ \bar{T}_{i,3}^j \succ \bar{T}_{i,1}^j & \bar{T}_{i,4}^j: \bar{a}_{i,4}^j \succ \bar{a}_{i,1}^j \succ \bar{a}_{i,6}^j \succ \bar{a}_{i,2}^j \succ \bar{a}_{i,3}^j \succ \bar{a}_{i,5}^j \quad C_1^{\bar{T}_{i,4}^j} = \{\bar{a}_{i,1}^j, \bar{a}_{i,2}^j, \bar{a}_{i,3}^j\}, C_2^{\bar{T}_{i,4}^j} = \{\bar{a}_{i,3}^j, \bar{a}_{i,4}^j, \bar{a}_{i,5}^j\} \\
& Q(\bar{T}_{i,4}^j) = 2, q^+(C_1^{\bar{T}_{i,4}^j}) = 1, q^+(C_2^{\bar{T}_{i,4}^j}) = 1
\end{array}$$

⁸ It can be verified that if $a_{i,1}^j$ is matched to some institute in $\Gamma(a_{i,1}^j)$, the above assignment is the only possibility that no blocking group arises.

The above instance $\bar{\mathcal{T}}_i^j$ has the following features, every one of which is crucial in our construction.

Feature A $\bar{\mathcal{T}}_i^j$ does not allow a stable matching; more importantly, if in the given matching μ , $|\mu(\bar{I}_{i,1}^j)| < 2$, then there exists at least one blocking group which is *not* of the form $(\bar{I}_{i,1}^j; \bar{a}_{i,x}^j, \bar{a}_{i,y}^j)$, where $\{\bar{a}_{i,x}^j, \bar{a}_{i,y}^j\} \subset \{\bar{a}_{i,t}^j\}_{t=1}^6$.

Feature B When the institute $\bar{I}_{i,1}^j$ is “removed” from the instance $\bar{\mathcal{T}}_i^j$ (i.e. $\bar{I}_{i,1}^j$ is struck from the preferences of the applicants $\{\bar{a}_{i,t}^j\}_{t=1}^6$), there exists at least one stable matching. For instance, the following matching μ is stable, if $\bar{I}_{i,1}^j$ is removed: $\mu(\bar{I}_{i,2}^j) = (\bar{a}_{i,6}^j, \bar{a}_{i,3}^j)$, $\mu(\bar{I}_{i,3}^j) = (\bar{a}_{i,5}^j, \bar{a}_{i,2}^j)$, $\mu(\bar{I}_{i,4}^j) = (\bar{a}_{i,4}^j, \bar{a}_{i,1}^j)$.

Feature C Each of the three institutes $\{\bar{I}_{i,t}^j\}_{t=2}^4$ uses exactly two intersecting classes; more importantly, institute $\bar{I}_{i,1}^j$ does not use any classification at all.

Our idea is to let the institute I_i^j in the gadget $\mathcal{Y}_{i,i'}$ “play the role” of $\bar{I}_{i,1}^j$ in the gadget of $\bar{\mathcal{T}}_i^j$ and add the other members in $\bar{\mathcal{T}}_i^j$ into gadget $\mathcal{Y}_{i,i'}$. To be precise, let $\Psi(I_i^j) = \bar{a}_{i,5}^j \succ \bar{a}_{i,2}^j \succ \bar{a}_{i,4}^j \succ \bar{a}_{i,6}^j \succ \bar{a}_{i,3}^j \succ \bar{a}_{i,1}^j$ (the same preference list of $\bar{I}_{i,1}^j$) and add the other members of $\bar{\mathcal{T}}_i^j$ into the gadget $\mathcal{Y}_{i,i'}$ without modifying their preference lists (and classifications). We now explain how the above features of $\bar{\mathcal{T}}_i^j$ help to realize the supposition in Claim *B* about the potential assignments of the applicants $\{a_{i,t}^j, a_{i',t}^j\}_{t=1}^2$ in a stable matching.

1. In a matching μ^ϕ , suppose that institute I_i^j is only assigned $a_{i,1}^j$ while $a_{i',1}^j$ is assigned to some institutes in $\Gamma(a_{i',1}^j)$ (the problematic case we discussed above). As a result, institute I_i^j can take one more applicant from the set $\{\bar{a}_{i,t}^j\}_{t=1}^6$. By Feature *A*, there must exist a blocking group involving the members in $\bar{\mathcal{T}}_i^j$. More importantly, this blocking group need not be composed of I_i^j and *two* applicants from $\{\bar{a}_{i,t}^j\}_{t=1}^6$.
2. In a matching μ^ϕ , suppose that institutes I_i^j is assigned two applicants from the set $\{a_{i,t}^j, a_{i',t}^j\}_{t=1}^2$. Then $\bar{I}_{i,1}^j$ can be regarded as being removed from the instance $\bar{\mathcal{T}}_i^j$. And there exists a stable matching among the other members of the instance $\bar{\mathcal{T}}_i^j$. This explains the necessity of Feature *B*.
3. Finally, since I_i^j already uses two intersecting classes, $\bar{I}_{i,1}^j$ should not use any more classes. This explains the reason why Feature *C* is necessary.

We have left the functions $\Gamma(a_{i,1}^j)$ and $\Gamma(a_{i',1}^j)$ unexplained so far. They contain institutes belonging to the clause gadgets, which will be the final component in our construction.

Clause Gadget Suppose that $c_j = x_1^j \vee x_2^j \vee x_3^j$. We create a clause gadget $\hat{\mathcal{Y}}_j$ composed of two institutes $\{\hat{I}_t^j\}_{t=1}^2$ and six applicants $\{\hat{a}_t^j\}_{t=1}^6$. Their preferences and classifications are summarized below.

We now explain how the Λ functions in the clause gadgets interact with the Γ functions in the literal-pair gadgets. The former is composed of applicants in the literal-pair gadgets while the latter is composed of institutes in the clause gadgets. Our intuition is that the only possible stable matchings in the clause gadgets will enforce exactly one of its three literals to be activated. To be precise, let $\pi(X)$ denote an arbitrary order among the elements in the set X . Then:

$$\begin{array}{l|l}
\hat{a}_1^j: \hat{I}_2^j \succ \hat{I}_1^j & \hat{I}_1^j: \hat{a}_5^j \succ \hat{a}_1^j \succ \hat{a}_2^j \succ \Lambda(x_1^j) \succ \hat{a}_6^j \succ \Lambda(x_2^j) \succ \hat{a}_3^j \succ \Lambda(x_3^j) \succ \hat{a}_4^j \\
\hat{a}_2^j: \hat{I}_1^j \succ \hat{I}_2^j & C_1^{j1} = \{\hat{a}_1^j, \hat{a}_2^j, \hat{a}_5^j, \hat{a}_6^j, \Lambda(x_1^j)\}, C_2^{j1} = \{\hat{a}_2^j, \hat{a}_5^j, \hat{a}_6^j, \Lambda(x_2^j)\} \\
\hat{a}_3^j: \hat{I}_2^j \succ \hat{I}_1^j & Q(\hat{I}_1^j) = 3, q^+(C_1^{j1}) = 2, q^+(C_2^{j1}) = 2 \\
\hat{a}_4^j: \hat{I}_1^j \succ \hat{I}_2^j & \hat{I}_2^j: \hat{a}_6^j \succ \hat{a}_2^j \succ \hat{a}_1^j \succ \hat{a}_5^j \succ \hat{a}_4^j \succ \hat{a}_3^j \\
\hat{a}_5^j: \hat{I}_2^j \succ \hat{I}_1^j & C_1^{j2} = \{\hat{a}_1^j, \hat{a}_2^j, \hat{a}_5^j, \hat{a}_6^j\}, C_2^{j2} = \{\hat{a}_1^j, \hat{a}_4^j, \hat{a}_6^j\}. \\
\hat{a}_6^j: \hat{I}_1^j \succ \hat{I}_2^j & Q(\hat{I}_2^j) = 3, q^+(C_1^{j2}) = 2, q^+(C_2^{j2}) = 2
\end{array}$$

$$\begin{aligned}
\Lambda(x_t^j) &= \pi(\{a_{i,1}^{j'} \mid \text{if } a_{i,1}^{j'} \in \mathcal{Y}_{i,i'}^{j'} \text{ and } x_i^{j'} = x_t^j\}), \text{ for } t \in \{1, 2, 3\}. \\
\Gamma(a_{i,1}^j) &= \pi(\{\hat{I}_1^{j'} \mid \text{if clause } c_{j'} = x_1^{j'} \vee x_2^{j'} \vee x_3^{j'} \text{ contains the literal } x_i^{j'}\})
\end{aligned}$$

Claim C Suppose there exists a stable matching μ in the derived instance. Then there can be only three possible outcomes for the members in the clause gadget $\hat{\mathcal{Y}}_j$.

1. $\mu(\hat{I}_1^j) = (\hat{a}_5^j, \hat{a}_1^j, \hat{a}_3^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_6^j, \hat{a}_2^j, \hat{a}_4^j)$. Moreover, in this case, x_2^j must be activated while x_1^j and x_3^j must remain deactivated.
2. $\mu(\hat{I}_1^j) = (\hat{a}_5^j, \hat{a}_2^j, \hat{a}_4^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_6^j, \hat{a}_1^j, \hat{a}_3^j)$. Moreover, in this case, x_3^j must be activated while x_1^j and x_2^j must remain deactivated.
3. $\mu(\hat{I}_1^j) = (\hat{a}_2^j, \hat{a}_6^j, \hat{a}_3^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_1^j, \hat{a}_5^j, \hat{a}_4^j)$. Moreover, in this case, x_1^j must be activated while x_2^j and x_3^j must remain deactivated.

Proof of Claim C. We have argued previously that in a stable matching μ , the applicants in the literal-pair gadgets cannot be matched to the institutes in the clause gadget. So both \hat{I}_1^j and \hat{I}_2^j can only have applicants from the set $\{\hat{a}_t^j\}_{t=1}^6$ in μ . We first make the following observation.

In a stable matching μ , one of the applicants in the following three pairs $\{\hat{a}_1^j, \hat{a}_2^j\}$, $\{\hat{a}_3^j, \hat{a}_4^j\}$, $\{\hat{a}_5^j, \hat{a}_6^j\}$ must be assigned to \hat{I}_1^j ; while the other applicant in the above three pairs must be assigned to \hat{I}_2^j .

The above observation can be easily verified. So there are eight possible combinations in μ . Except the three combinations listed in the claim, the other five will cause blocking groups to arise.

1. $\mu(\hat{I}_1^j) = (\hat{a}_1^j, \hat{a}_6^j, \hat{a}_3^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_2^j, \hat{a}_5^j, \hat{a}_4^j)$. Then $(\hat{I}_1^j; \mu(\hat{I}_1^j) | \hat{a}_2^j \hat{a}_6^j)$ blocks μ .
2. $\mu(\hat{I}_1^j) = (\hat{a}_1^j, \hat{a}_6^j, \hat{a}_4^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_2^j, \hat{a}_5^j, \hat{a}_3^j)$. Then again $(\hat{I}_1^j; \mu(\hat{I}_1^j) | \hat{a}_2^j \hat{a}_6^j)$ blocks μ .
3. $\mu(\hat{I}_1^j) = (\hat{a}_5^j, \hat{a}_2^j, \hat{a}_3^j)$. Note that the tuple $\mu(\hat{I}_2^j) = (\hat{a}_6^j, \hat{a}_1^j, \hat{a}_4^j)$ is not feasible. So $\mu(\hat{I}_2^j) = (\hat{a}_6^j, \hat{a}_1^j)$. But then $(\hat{I}_2^j; \mu(\hat{I}_2^j) | \hat{a}_3^j)$ blocks μ .
4. $\mu(\hat{I}_1^j) = (\hat{a}_5^j, \hat{a}_1^j, \hat{a}_4^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_6^j, \hat{a}_2^j, \hat{a}_3^j)$. In this case, institute \hat{I}_1^j can replace \hat{a}_4^j with an applicant in $\Lambda(x_2^j)$ or an applicant in $\Lambda(x_3^j)$ to have a blocking group, unless both x_2^j and x_3^j are activated. However, this is impossible, since we have a literal-pair gadget $\mathcal{Y}_{2,3}^j$ and by Claim B, at most one of x_2^j and x_3^j can be activated. So we have a contradiction.
5. $\mu(\hat{I}_1^j) = (\hat{a}_2^j, \hat{a}_6^j, \hat{a}_4^j)$, $\mu(\hat{I}_2^j) = (\hat{a}_1^j, \hat{a}_5^j, \hat{a}_3^j)$. Then the existence of a blocking group involving \hat{I}_1^j and some applicants in $\Lambda(x_1^j) \cup \Lambda(x_3^j)$ is guaranteed. The reason is similar to the previous case: x_1^j and x_3^j cannot be both activated.

Finally, we remark that the three possible outcomes in μ listed in the lemma will guarantee that exactly one of the three literals in clause c_j can be activated. The reason is again the same as in the last two cases that we just explained. This completes the proof of Claim C . \square

Now by Claim C , we establish Theorem 16

C Missing Proofs of Section 4

Lemma 17. *In $LCSM$, if there is no lower bound, i.e., given any class C_j^i , $q^-(C_j^i) = 0$, then a stable matching as defined in Definition 2 can be equivalently defined as follows. A feasible matching μ is stable if and only if there is no blocking pair. A pair (i, a) is blocking, given that $\mu(i) = (a_{i1}, a_{i2}, \dots, a_{ik})$, $k \leq Q(i)$, if*

- $i \succ_a \mu(a)$;
- for any class $C_{at}^i \in a(\mathbb{C}(i))$, $|\mathbb{L}_{\succ_a}^i \cap \mu(i) \cap C_{at}^i| < q^+(C_{at}^i)$.

Proof. If we have a blocking group $(i; g)$, institute i and the highest ranking applicant in $g \setminus \mu(i)$ must be a blocking pair. Conversely, given a blocking pair $(i; a)$, assuming that $|\mu(i)| = Q(i)$ (the case that $|\mu(i)| < Q(i)$ follows a similar argument), we can form a blocking group $(i; \mu(i) |^{a^\dagger} a)$, where a^\dagger is chosen as follows: (1) if there exists a class $C_{at}^i \in a(\mathbb{C}(i))$ such that $|\mu(i) \cap C_{at}^i| = q^+(C_{at}^i)$, choose the smallest such class $C_{at}^i \in a(\mathbb{C}(i))$ and let a^\dagger be the lowest ranking applicant in $\mu(i) \cap C_{at}^i$; (2) otherwise, a^\dagger is simply the lowest ranking applicant in $\mu(i)$. \square

Lemma 19. *Every stable matching solution x satisfies the comb inequality for any comb $K(i, S(A^i))$:*

$$x(K(i, S(A^i))) \equiv x(S(A^i)) + \sum_{a_j \in A^i} x(T(i, a_j) \setminus \{i, a_j\}) \geq |A^i|.$$

We use the following notation to facilitate the proof. Give a tuple A^i , we define y_{ia} as follows:

$$y_{ia} = \begin{cases} 1 & \text{either } a \in A^i, x(T(i, a)) = 1; \text{ or } a \notin A^i, x_{ia} = 1, \text{ and } (i, a) \in S(A^i); \\ 0 & \text{o.w.} \end{cases}$$

Let $y(C_j^i) = \sum_{a \in \mathbb{L}^i \cap C_j^i} y_{ia}$. This quantity indicates how much a class C_j^i contributes to the comb value $x(K(i, S(A^i)))$. Thus, if U is a set of classes in $\mathbb{C}(i)$ partitioning \mathbb{L}^i , then $x(K(i, S(A^i))) = \sum_{C_j^i \in U} y(C_j^i)$.

Proof. We prove by showing that if $x(K(i, S(A^i))) < |A^i|$, there exists a blocking pair (i, a^\dagger) , where $a^\dagger \in A^i$. We proceed by contradiction. First note that there exists a non-empty subset $G \subseteq A^i$ of applicants a for whom $x(T(i, a)) = 0$, otherwise, $x(K(i, S(A^i))) \geq |A^i|$, an immediate contradiction. For each applicant $a \in G$, there must exist a class $C_{al}^i \in a(\mathbb{C}(i))$ for which $\sum_{a' \in \mathbb{L}_{\succ_a}^i \cap C_{al}^i} x_{ia'} = q^+(C_{al}^i)$, otherwise, (i, a) is a blocking pair and we are done. Now for each applicant $a \in G$, choose the smallest class C_{al}^i for which $\sum_{a' \in \mathbb{L}_{\succ_a}^i \cap C_{al}^i} x_{ia'} = q^+(C_{al}^i)$ and denote this class as \overline{C}_a . We introduce a procedure to organize a set U of disjoint classes.

Let G be composed of $a_1, a_2, \dots, a_{|G|}$ ordered based on their decreasing rankings on \mathbb{L}^i

For $i = 1$ **To** $|G|$

if $a_i \in C \in U$, **then** do nothing

else $U := U \setminus \{C \mid C \in U, C \subset \overline{C}_{a_i}\} // \overline{C}_{a_i}$ may be a superclass of some classes in U

$U := U \cup \{\overline{C}_{a_i}\}$. // adding \overline{C}_{a_i} into U

Claim The output U from the above procedure comprises of a disjoint set of classes containing all applicants in G , and for each class $C_j^i \in U$, $y(C_j^i) \geq q^+(C_j^i)$.

We will prove the claim shortly. Now

$$x(K(i, S(A^i))) = \sum_{C_j^i \in U} y(C_j^i) + |A^i \setminus \{\cup_{C_j^i \in U} C_j^i\}| \geq \sum_{C_j^i \in U} q^+(C_j^i) + |A^i \setminus \{\cup_{C_j^i \in U} C_j^i\}| \geq |A^i|,$$

a contradiction. \square

Proof of the Claim. It is easy to see that the classes in U are disjoint and contain all applicants in G . Below we show that during the execution of the procedure, if $C_j^i \in U$, then $y(C_j^i) \geq q^+(C_j^i)$. We proceed by induction on the number of times U is updated. In the base case U is an empty set so there is nothing to prove.

For the induction step, assume that a_l is being examined and \bar{C}_{a_l} is about to be added into U . Observe that even though $\sum_{a \in \mathbb{L}_{>a_l}^i \cap \bar{C}_{a_l}} x_{ia} = q^+(\bar{C}_{a_l})$, there is no guarantee that if $x_{ia} = 1$, then $y_{ia} = 1$ for each $a \in \mathbb{L}_{>a_l}^i \cap \bar{C}_{a_l}$. The reason is that there may exist some class $C_j^i \in a(\mathbb{C}(i))$ for which $|A^i \cap C_j^i \cap \mathbb{L}_{>a_l}^i| = q^+(C_j^i)$ and $a \notin A^i$. Then (i, a) is not part of the shaft $x(S(A^i))$ and $y_{ia} = 0$.

To deal with the above situation, we need to do some case analysis. Let B be the set of subclasses C_j^i of \bar{C}_{a_l} for which $|A^i \cap C_j^i \cap \mathbb{L}_{>a_l}^i| = q^+(C_j^i)$. Choose D to be the subclasses of \bar{C}_{a_l} so that $\mathfrak{R}(B \cup U) \cup D$ partitions \bar{C}_{a_l} . We make three observations below.

- (i) for each class $C_j^i \in \mathfrak{R}(B \cup U)$ and $C_j^i \in U$, $y(C_j^i) \geq q^+(C_j^i) \geq \sum_{a \in \mathbb{L}_{>a_l}^i \cap C_j^i} x_{ia}$.
- (ii) for each class $C_j^i \in D$, if $a \in \mathbb{L}_{>a_l}^i \cap C_j^i$ and $x_{ia} = 1$, then $y_{ia} = 1$.
- (iii) for each class $C_j^i \in \mathfrak{R}(B \cup U)$ and $C_j^i \notin U$, then for each applicant $a \in \mathbb{L}_{>a_l}^i \cap C_j^i \cap A^i$, either $a \in G$ and $a \in C \in U$, or that $a \notin G$ (implying that $x(T(i, a)) = 1$). Moreover, $y(C_j^i) \geq \sum_{a \in \mathbb{L}_{>a_l}^i \cap C_j^i} x_{ia}$

(i) is because of the induction hypothesis and the feasibility assumption of x . (ii) follows from the fact that a ranks higher than a_l and the way we define a class in D . For (iii), first notice that if $C_j^i \in \mathfrak{R}(B \cup U)$ and $C_j^i \notin U$, then such a class C_j^i must be part of $\mathfrak{R}(B)$ and C_j^i may contain some classes in U . Now suppose that $a_i \in G \cap \mathbb{L}_{>a_l}^i$ but does not belong to any class in U . Then our procedure would have added the class \bar{C}_{a_i} into U before examining a_l , a contradiction. To see the last statement of (iii), let G' be set of applicants in $\mathbb{L}_{>a_l}^i \cap C_j^i \cap A^i$ who do not belong to any classes in U . Then

$$y(C_j^i) \geq \sum_{C_k^i \in U, C_k^i \subset C_j^i} y(C_k^i) + |G'| \geq \sum_{C_k^i \in U, C_k^i \subset C_j^i} q^+(C_k^i) + |G'| \geq q^+(C_j^i) \geq \sum_{a \in \mathbb{L}_{>a_l}^i \cap C_j^i} x_{ia},$$

where the first inequality follows from the first part of (iii), the second inequality the induction hypothesis, the third the fact that $C_j^i \in \mathfrak{R}(B)$ (thus $|\mathbb{L}_{>a_l}^i \cap C_j^i \cap A^i| = q^+(C_j^i)$), and the fourth the feasibility assumption of x .

Now combining all the three observations, we conclude that

$$y(\bar{C}_{a_l}) = \sum_{C_j^i \in \mathfrak{R}(B \cup U)} y(C_j^i) + \sum_{C_k^i \in D} y(C_k^i) \geq \sum_{C_k^i \in \mathfrak{R}(B_l \cup U) \cup D} \sum_{a \in \mathbb{L}_{>a_l}^i \cap C_k^i} x_{ia} = q^+(\bar{C}_{a_l}),$$

and the induction step is completed. \square

Lemma 20. *Every stable matching solution x satisfies the following inequality for any class-tuple t_j^i :*

$$\sum_{a_{ij} \in t_j^i} x(T(i, a_{ij}) \setminus \{i, a_{ij}\}) \geq \sum_{a \in C_j^i \cap \mathbb{L}_{\prec t_j^i}^i} x_{ia} \quad (*)$$

Proof. We prove by contradiction. Suppose that in a given class-tuple t_j^i (*) does not hold. We will show that we can find a blocking pair (i, a^\dagger) , where $a^\dagger \in t_j^i$. Let the set of applicants $a \in t_j^i$ with $x(T(i, a)) = 0$ be G , $\alpha = \sum_{a' \in \mathbb{L}_{\prec t_j^i}^i \cap C_j^i} x_{ia'} > 0$, and $\beta = \sum_{a' \in t_j^i} x_{ia'}$. By assumption, at most $\alpha - 1$ applicants $a \in t_j^i$ have $x(T(i, a) \setminus \{(i, a)\}) = 1$. Thus,

$$|G| \geq q^+(C_j^i) - \beta - \alpha + 1. \quad (11)$$

Claim: At least one applicant $a^\dagger \in G$ belongs to a sequence of classes $C_{a^\dagger t}^i \in a^\dagger(\mathbb{C}(i))$ such that if $C_{a^\dagger t}^i \subseteq C_j^i$, then $\sum_{a' \in \mathbb{L}_{\succ a^\dagger}^i \cap C_{a^\dagger t}^i} x_{ia'} < q^+(C_{a^\dagger t}^i)$.

We will prove the claim shortly. Observe that given any class $C_k^i \supset C_j^i$, $\sum_{a' \in \mathbb{L}_{\succ a^\dagger}^i \cap C_k^i} x_{ia'} < q^+(C_k^i)$: as $\alpha > 0$, some applicant $a^\phi \in C_k^i$ ranking lower than a^\dagger has $x_{ia^\phi} = 1$ and Constraint (2) enforces that $\sum_{a' \in \mathbb{L}^i \cap C_k^i} x_{ia'} \leq q^+(C_k^i)$. Combining the above facts, we conclude that (i, a^\dagger) is a blocking pair. \square

Proof of the Claim. We prove by contradiction. Suppose that for every applicant $a \in G$, there exists some class $C_{at}^i \in a(\mathbb{C}(i))$, $C_{at}^i \subseteq C_j^i$, and $\sum_{a' \in \mathbb{L}_{\succ a}^i \cap C_{at}^i} x_{ia'} = q^+(C_{at}^i)$.

Let B be the set of classes $C_k^i \subseteq C_j^i$ such that C_k^i contains an applicant $a \in G$ and $\sum_{a' \in \mathbb{L}_{\succ a}^i \cap C_k^i} x_{ia'} = q^+(C_k^i)$ (which then will equal $\sum_{a' \in \mathbb{L}_{\succeq t_j^i}^i \cap C_k^i} x_{ia'}$ due to Constraint (2)). For each class $C_k^i \in \mathfrak{R}(B)$,

$$\sum_{a \in \mathbb{L}_{\succeq t_j^i}^i \cap C_k^i} x_{ia} = q^+(C_k^i) \geq |t_j^i \cap C_k^i| = \sum_{a' \in \mathbb{L}_{\succeq t_j^i}^i \cap t_j^i \cap C_k^i} x_{ia'} + |G \cap C_k^i|, \quad (12)$$

where the first inequality follows from the definition of the class-tuple. Now we have

$$q^+(C_j^i) - \alpha - \beta \geq \sum_{C_k^i \in \mathfrak{R}(B)} \sum_{a' \in (\mathbb{L}_{\succ a^\dagger}^i \cap C_k^i) \setminus t_j^i} x_{ia'} \geq \sum_{C_k^i \in \mathfrak{R}(B)} |G \cap C_k^i| = |G| \geq q^+(C_j^i) - \alpha - \beta + 1,$$

a contradiction. Note that the first inequality follows from Constraint (2), the second inequality from (12), the equality right after is because every applicant in G belongs to some class in B , and the last inequality is due to (11). \square

D Separation Oracle in Section 4.1

It is clear that Constraints (1)(2)(4) can be separated in polynomial time. So we assume that x satisfies these constraints and focus on finding a violated Constraint (3) and/or Constraint (5).

Separating Constraint (3) We first make an observation. For each institute i , it suffices to check whether all the combs with exactly $Q(i)$ teeth satisfy Constraint (3). To see this, suppose that there is a feasible tuple \bar{A}^i with less than $Q(i)$ applicants and $x(K(i, S(\bar{A}^i))) < |\bar{A}^i|$. Then we can add suitable applicants into \bar{A}^i to get a feasible tuple A^i with exactly $Q(i)$ applicants. Noticing that $x(S(A^i)) \leq x(S(\bar{A}^i))$, we have

$$\begin{aligned} x(K(i, S(A^i))) &\leq x(S(\bar{A}^i)) + \sum_{a \in \bar{A}^i} x(T(i, a) \setminus \{(i, a)\}) + \sum_{a \in A^i \setminus \bar{A}^i} x(T(i, a) \setminus \{(i, a)\}) \\ &< |\bar{A}^i| + \sum_{a \in A^i \setminus \bar{A}^i} x(T(i, a) \setminus \{(i, a)\}) \\ &\leq |\bar{A}^i| + |A^i| - |\bar{A}^i| \\ &= |A^i|, \end{aligned}$$

where the last inequality follows from our assumption that x satisfies Constraint (1).

To illustrate our idea, we first explain how to deal with the case that the original classification $\mathbb{C}(i)$ is just a partition over \mathbb{L}^i (before we add the pseudo root class $C_{\#}^i$). We want to find out the tuple A^i of length $Q(i)$, whose lowest ranking applicant is a^\dagger , which gives the smallest $x(K(i, S(A^i)))$. If we have this information for all possible a^\dagger , we are done. Note that because of our previous discussion, if there is no feasible tuple of length $Q(i)$ whose lowest ranking applicant is a^\dagger , we can ignore those cases.

Our main idea is to decompose the value of $x(K(i, S(A^i)))$ based on the classes and use dynamic programming to find out the combinations of the tooth-applicants that give the smallest comb values. More precisely,

Definition 27. Assume that $A_j^i \subseteq C_j^i$, $0 \leq |A_j^i| \leq q^+(C_j^i)$, and all applicants in A_j^i rank higher than a^\dagger . Let

$$\begin{aligned} x(A_j^i, a^\dagger) &= \sum_{a \in \mathbb{L}_{>a^\dagger}^i \cap C_j^i, (i, a) \in S(A_j^i)} x_{ia} + \sum_{a \in A_j^i} x(T(i, a) \setminus \{(i, a)\}) \\ Z(C_j^i, s_j, a^\dagger) &= \min_{A_j^i: A_j^i \subseteq C_j^i, |A_j^i|=s_j} x(A_j^i, a^\dagger). \end{aligned}$$

Note that this definition requires that if x_{ia} contributes to $x(A_j^i, a^\dagger)$, then a has to rank higher than a^\dagger , belongs to C_j^i , and the (i, a) is part of the shaft $S(A_j^i)$.

Suppose that we have properly stored all the possible values of $Z(C_j^i, s_j, a^\dagger)$ and assume that $a^\dagger \in C_{j'}^i$. Then for each class $C_j^i \neq C_{j'}^i$, assume that $0 \leq s_j \leq q^+(C_j^i)$ and for class $C_{j'}^i$, $0 \leq s_{j'} \leq q^+(C_{j'}^i) - 1$, then the tuple A^i whose lowest ranking applicant is a^\dagger , that gives the smallest comb value is the following one:

$$x(K(i, S(A^i))) = x(T(i, a^\dagger)) + \min_{s_j: \sum_{C_j^i \in \mathbb{C}(i)} s_j = Q(i) - 1} \sum_{C_j^i \in \mathbb{C}(i)} Z(C_j^i, s_j, a^\dagger).$$

The above quantity can be calculated using standard dynamic programming technique. So the question boils down to how to calculate $Z(C_j^i, s_j, a^\dagger)$. There are two cases.

- Suppose that $s_j < q^+(C_j^i)$. Observe that all the positive x_{ia} , where $a \in C_j^i \cap \mathbb{L}_{>a^\dagger}^i$ will contribute to the value $x(A_j^i, a^\dagger)$. So, to calculate $Z(C_j^i, s_j, a^\dagger)$, we only need to find out the s_j applicants a in $C_j^i \cap \mathbb{L}_{>a^\dagger}^i$ with the smallest values $x(T(i, a) \setminus \{(i, a)\})$. This can be easily done in polynomial time.
- Suppose that $s_j = q^+(C_j^i)$. Different from the previous case, some of the positive x_{ia} , where $a \in C_j^i \cap \mathbb{L}_{>a^\dagger}^i$ will not contribute to $x(A_j^i, a^\dagger)$. Our idea is to “pin down” the lowest ranking applicant $a^\ddagger \in A_j^i$. After we pin down a^\ddagger , we know that the positive x_{ia} s that contribute to $x(A_j^i, a^\dagger)$ are those with $a \in C_j^i \cap \mathbb{L}_{>a^\ddagger}^i$, while those with $a \in C_j^i \cap (\mathbb{L}_{<a^\ddagger}^i \cap \mathbb{L}_{>a^\dagger}^i)$ do not. So what remains to be done is to find out the $s_j - 1$ applicants a in $C_j^i \cap \mathbb{L}_{>a^\ddagger}^i$ with the smallest values $x(T(i, a) \setminus \{(i, a)\})$. We then enumerate all possible $a^\ddagger \in C_j^i \cap \mathbb{L}_{>a^\dagger}^i$ and find out the applicant a^\ddagger that gives the smallest $Z(C_j^i, s_j, a^\dagger)$. The whole process can be done in polynomial time.

We now explain how to generalize to the case that $\mathbb{C}(i)$ is a laminar family. In the previous simplified case that $\mathbb{C}(i)$ is a partition, we critically collect the following information:

Suppose that $A_j^i \subseteq C_j^i$ and $0 \leq |A_j^i| \leq q^+(C_j^i)$ and all applicants in A_j^i rank at least as high as a^\ddagger and strictly higher than a^\dagger . What is the choice of A_j^i so that $x(S(A_j^i), a^\dagger) + \sum_{a \in A_j^i} x(T(i, a) \setminus \{(i, a)\})$ is minimized?

The above question motivates the following definition.

Definition 28. Assume that $A_j^i \subseteq C_j^i$, $0 \leq |A_j^i| \leq q^+(C_j^i)$ and all applicants in A_j^i rank at least as high as a^\ddagger , who in turn, ranks higher than a^\dagger . Furthermore, let $x(A_j^i, a^\dagger)$ inherit the definition as defined in Definition 27. We define

$$Z(C_j^i, s_j, a^\ddagger, a^\dagger) = \min_{A_j^i: A_j^i \subseteq C_j^i, |A_j^i|=s_j, \text{all applicants in } A_j^i \text{ rank at least as high as } a^\ddagger} x(A_j^i, a^\dagger).$$

For the case that $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ is not well-defined, e.g., there is no feasible tuple of length s_j so that all applicants in $A_j^i \subseteq C_j^i$ rank at least as high as a^\ddagger , or $a^\dagger \succeq_i a^\ddagger$, let $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ be an arbitrary large value.

In the following, we also associate each $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ with the corresponding feasible tuple A_j^i that realizes this value⁹. This helps us to identify the tuple that is violated in Constraint (3).

Theorem 29. For all classes $C_j^i \in \mathbb{C}(i)$, we can correctly calculate all possible $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ in polynomial time, for all a^\dagger and a^\ddagger combinations.

Proof. We first remark that in the calculation, we indeed only need to find out all the values of $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ for $a^\ddagger \in C_j^i$. For the case of $a^\ddagger \notin C_j^i$, simply copy the value of $Z(C_j^i, s_j, \bar{a}^\ddagger, a^\dagger)$ where $\bar{a}^\ddagger \in C_j^i$ and is the lowest ranking applicant that ranks higher than a^\ddagger . If there is no such applicant, let $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ be an arbitrary large value.

We now give an inductive proof based on the height of class C_j^i in the tree structure of $\mathbb{C}(i)$. The base case is when C_j^i is a leaf class. Then the calculation of all $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ can be done in essentially the same way as we have shown in the case that $\mathbb{C}(i)$ is a partition.

⁹ If $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ is some arbitrary large value, let its corresponding tuple be ϕ and indeed this will not matter.

For the induction step, let C_j^i be a non-leaf class and assume that $a^\ddagger \in C_{k'}^i \in c(C_j^i)$. To calculate $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$, we need to find out a feasible tuple A_j^i of size s_j , all of whose applicants rank at least as high as a^\ddagger so that $x(A_j^i, a^\dagger)$ is minimized.

Observe that a feasible tuple A_j^i can be decomposed into a set of tuples $A_j^i = \bigcup_{C_k^i \in c(C_j^i)} A_k^i$, where $A_k^i \subseteq C_k^i \in c(C_j^i)$.

1. Suppose that $s_j < q^+(C_j^i)$. Then by definition, $x(S(A_j^i), a^\dagger) = \sum_{C_k^i \in c(C_j^i)} x(S(A_k^i), a^\dagger)$. So

$$x(A_j^i, a^\dagger) = \sum_{C_k^i \in c(C_j^i)} \left[\sum_{a \in A_k^i} x(T(i, a) \setminus \{(i, a)\}) + x(S(A_k^i), a^\dagger) \right].$$

For each class $C_k^i \in c(C_j^i)$, the minimum quantity $\sum_{a \in A_k^i} x(T(i, a) \setminus \{(i, a)\}) + x(S(A_k^i), a^\dagger)$ is exactly $Z(C_k^i, s_k, a^\ddagger, a^\dagger)$. As a result, for each class $C_k^i \neq C_{k'}^i$, let $0 \leq s_k \leq q^+(C_k^i)$, and for class $C_{k'}^i$, let $0 \leq s_{k'} \leq q^+(C_{k'}^i) - 1$:

$$Z(C_j^i, s_j, a^\ddagger, a^\dagger) = x(T(i, a^\ddagger)) + \min_{s_k: \sum s_k = s_j - 1} \sum_{C_k^i \in c(C_j^i)} Z(C_k^i, s_k, a^\ddagger, a^\dagger).$$

Thus, we can find out $Z(C_j^i, s_j, a^\ddagger, a^\dagger)$ by dynamic programming.

2. Suppose that $s_j = q^+(C_j^i)$. Note that this time since the class C_j^i will be ‘‘saturated’’, the term $x(S(A_j^i), a^\dagger)$ does not get any positive values x_{ia} , provided that $a \in C_j^i \cap (\mathbb{L}_{>a^\dagger}^i \cap \mathbb{L}_{<a^\dagger}^i)$. So $x(S(A_j^i), a^\dagger) = \sum_{C_k^i \in c(C_j^i)} x(S(A_k^i), a^\dagger)$ and this implies that

$$x(A_j^i, a^\dagger) = \sum_{C_k^i \in c(C_j^i)} \left[\sum_{a \in A_k^i} x(T(i, a) \setminus \{(i, a)\}) + x(S(A_k^i), a^\dagger) \right].$$

Let \bar{a}^\ddagger be the lowest ranking applicant that ranks higher than a^\ddagger . Then for each class, $C_k^i \in c(C_j^i)$, the minimum quantity $\sum_{a \in A_k^i} x(T(i, a) \setminus \{(i, a)\}) + x(S(A_k^i), \bar{a}^\ddagger)$ is exactly $Z(C_k^i, s_k, \bar{a}^\ddagger, a^\dagger)$. Assuming that for each class $C_k^i \neq C_{k'}^i$, let $0 \leq s_k \leq q^+(C_k^i)$, and let $0 \leq s_{k'} \leq q^+(C_{k'}^i) - 1$, we have

$$Z(C_j^i, s_j, a^\ddagger, a^\dagger) = x(T(i, a^\ddagger)) + \min_{s_k: \sum s_k = s_j - 1} \sum_{C_k^i \in c(C_j^i)} Z(C_k^i, s_k, \bar{a}^\ddagger, a^\dagger).$$

As before, this can be calculated by dynamic programming. □

Now choose the smallest $Z(C_{\#}^i, Q(i) - 1, a^\ddagger, a^\dagger)$ among all possible a^\ddagger who rank higher than a^\dagger and assume that $A_{\#}^i$ is the corresponding tuple. It is easy to see that among all feasible tuples A^i of length $Q(i)$ whose lowest ranking applicant is a^\dagger , the one has the smallest comb value $x(K(i, S(A^i)))$, is exactly the tuple $A_{\#}^i \cup \{a^\dagger\}$.

Separating Constraint (5) We again make use of dynamic programming. The idea is similar to the previous one and the task is much simpler, so we will be brief.

Suppose that we are checking all the class-tuples \mathbb{T}_j^i corresponding to class C_j^i . Let $T_{j, a^\dagger}^i \subseteq \mathbb{T}_j^i$ be the subset of class-tuples whose lowest ranking applicants is a^\dagger . We need to find out the class-tuple $t_{j, a^\dagger}^i \in T_{j, a^\dagger}^i$ with the smallest value

$$x(T(i, a^\dagger) \setminus \{(i, a^\dagger)\}) + \sum_{a \in t_{j, a^\dagger}^i \setminus \{a^\dagger\}} x(T(i, a) \setminus \{(i, a)\}),$$

and check whether this value is no less than $\sum_{a \in C_j^i \cap \mathbb{L}_{\prec a^\dagger}^i} x_{ia}$. If it is, then we are sure that all class-tuples in T_{j, a^\dagger}^i satisfy Constraint (5), otherwise, we find a violated constraint. The above quantity can be easily calculated by dynamic programming as before.

E A Counter Example for Section 4.2

The example shown in Figure 4 contains five stable matchings. If we apply the median choice operation on all of them, we get the stable matching μ_2 , which does not give institutes i_1 and i_2 their lexicographical median outcome.

Institute Preferences	Classifications	Class Bounds
$i_1: a_x a_y a_1 a_2 a_3 a_4$	$C_1^1 = \{a_1, a_2\}, C_2^1 = \{a_3, a_4\}$	$Q(i_1) = 2, q^+(C_1^1) = 1, q^+(C_2^1) = 1$
$i_2: a_z a_w a_2 a_1 a_4 a_3$	$C_1^2 = \{a_1, a_2\}, C_2^2 = \{a_3, a_4\}$	$Q(i_2) = 2, q^+(C_1^2) = 1, q^+(C_2^2) = 1$
$i_3: a_1 a_2 a_3 a_4 a_x a_y a_z a_w$		$Q(i_3) = 4$

Applicant Preferences
$a_1: i_2 i_1 i_3$
$a_2: i_1 i_2 i_3$
$a_3: i_2 i_1 i_3$
$a_4: i_1 i_2 i_3$
$a_x: i_3 i_1$
$a_y: i_3 i_1$
$a_z: i_3 i_2$
$a_w: i_3 i_2$

Stable Matchings
$\mu_1 = \{(i_1; a_x, a_y), (i_2; a_z, a_w), (i_3; a_1, a_2, a_3, a_4)\}$
$\mu_2 = \{(i_1; a_1, a_3), (i_2; a_2, a_4), (i_3; a_x, a_y, a_z, a_w)\}$
$\mu_3 = \{(i_1; a_1, a_4), (i_2; a_2, a_3), (i_3; a_x, a_y, a_z, a_w)\}$
$\mu_4 = \{(i_1; a_2, a_3), (i_2; a_1, a_4), (i_3; a_x, a_y, a_z, a_w)\}$
$\mu_5 = \{(i_1; a_2, a_4), (i_2; a_1, a_3), (i_3; a_x, a_y, a_z, a_w)\}$

Fig. 4. An example of median choice stable matching which does not give the institutes their lexicographically median outcome.
