Mechanism Design Notes

Clay Thomas
claytont@princeton.edu

April 15, 2019

1 Differed Acceptance and Stable Matching

Definition 1.1. *A matching market is a collection D of doctors and H of hospitals, where each doctor $d \in D$ has a ranking (strict total order) denoted $\succ_d$ over hospitals $h \in H$, and vice-versa.*

A matching is a one-to-one assignment of doctors to hospitals, which we denote by $\mu : D \cup H \to D \cup H \cup \{\emptyset\}$, such that all the reasonable conditions are met to make it a matching. We write $\mu(i) = \emptyset$ of agent i is unmatched.

A matching μ is stable *if for every unmatched doctor/hospital pair $(d, h) \in D \times H$, we do not simultaneously have $h \succ_d \mu(d)$ and $d \succ_h \mu(h)$.*

The doctor-proposing differed acceptance algorithm is then:

```
Let $U = D$ be the set of unmatched doctors
Let $\mu$ be an all empty matching
while $U \neq \emptyset$ and some $d \in U$ has not proposed to every hospital do
    Pick such a $d$ (in any order)
    $d$ "proposes" to their highest-ranked hospital $h$ which they have not yet proposed to
    if $\mu(h) = \emptyset$ then
        Set $\mu(h) = d$, remove $d$ from $U$
    if $\mu(h) = d_0$ and $d \succ_h d_0$ then
        Set $\mu(h) = d$, remove $d$ from $U$, add $d_0$ to $U$
    else $h$ remains matched to $d_0$ and $d$ remains in $U$
```

Claim 1.2. *The DA algorithm terminates.*

Proof. Every doctor will propose to every hospital at most once. □

Intuitively, this algorithm starts with the doctors doing whatever they prefer the most, then doing the minimal amount of work to make the matching stable. Indeed, doctors propose in their order of preference. If a hospital ever rejected a doctor they prefer, then remained with their current match, that pair would clearly create an instability in the final mechanism.

Claim 1.3. *The output of DA is a stable matching.*

Proof. Consider a pair $d \in D$, $h \in H$ which is unmatched in the output matching μ. Suppose for contradiction $h \succ_d \mu(d)$ and $d \succ_h \mu(h)$. Well, in the DA algorithm, d would propose to h before $\mu(d)$. However, it's easy to observe that once a hospital is proposed to, they remains matched and can only increase their preference for their match. This contradicts the fact that h was eventually matched to $\mu(h)$. □

Note that this algorithm gives us a very interesting existence result: it was not at all clear that stable matching existed before we had this algorithm.

Claim 1.4. *If a doctor $d \in D$ is ever rejected by a hospital $h \in H$ during some run of DA (that is, d proposes to h and h does not accept) then no stable matching can pair d to h.*

Proof. Let μ be any matching, not necessarily stable. We will show that if h rejects $\mu(h)$ at any step of DA, then μ is not stable.

Consider the first time during in the run of DA where such a rejection occurred. In particular, let h reject $d = \mu(h)$ in favor of $d' \neq d$ (either because d' proposed to h, or because d' was already matched to h and d proposed). We have $d' \succ_h d$. We have $\mu(d') \neq h$, simply because μ is a matching. Because this is the first time any doctor has been rejected by a match from μ, d' has not yet proposed to $\mu(d')$. This means $h \succ_{d'} \mu(d')$. However, this means μ is not stable.

Thus, no hospital can ever reject a stable partner in doctor-proposing DA. □

Corollary 1.5. *Let* $\text{best}(d)$ *denote the most preferred match d can achieve in any stable matching, i.e. the maximum according to $\succ_d$ of the set $\{h \in H : \exists \mu : \mu \text{ is stable and } \mu(d) = h\}$.*

In the match returned by DA, every $d \in D$ is paired to $\text{best}(d)$.

Corollary 1.6. *The matching output by the DA algorithm is independent of the order in which doctors are selected to propose.*

Corollary 1.7. *If a set of doctors $\bar{D}$ are rejected by every hospital during DA, then no stable matching will match any doctor in $\bar{D}$. Moreover, in every stable matching, the set of unmatched doctors is the same.*

Proof. Let $\bar{D}$ be unmatched in DA, and $\bar{D}'$ unmatched in some other stable matching μ. The doctors in $\bar{D}$ are rejected by every hospital, so $\bar{D} \subseteq \bar{D}'$. But these sets must be the same size, so they are equal (assuming every hospital prefers any doctor to being unmatched). □

This formalizes our intuition that DA moves the doctor's down their preference lists the minimal amount required to enforce stability.

A completely dual phenomenon occurs for the hospital's preferences:

Claim 1.8. *In the match returned by DA, every $h \in H$ is paired to their worst stable match in D.*

Proof. Let $d \in D$ and $h \in H$ be paired by doctor-proposing differed acceptance. Let μ be any stable matching which does not pair d and h. We must have $h >_d \mu(d)$, because $h = \text{best}(d)$. If $d >_h \mu(h)$, then μ is not stable. Thus, h cannot be stably matched to any doctor they prefer less than d. □

Claim 1.9. *The set of unmatched hospitals is the same in any stable matching.*

Proof. Hospital-proposing DA also produces a stable matching, so this simply follows from claim 1.7. □

To strengthen the power of our solution concept further, we can also compare stable matching to arbitrary matchings.

Claim 1.10. *Let μ be doctor-proposing DA matching, and let μ' be an arbitrary matching, $\mu \neq \mu'$. Then it is not possible for every doctor to prefer their match in μ' over their match in μ.*

Proof Sketch? . Suppose for contradiction that $\mu'(d) \geq_d \mu(d)$ for every $d \in D$. Then every doctor will propose to $\mu'(d)$ before $\mu(d)$. Now, because DA is independent of execution order, we can find an execution where every doctor d is tentatively accepted by $\mu'(d)$. In particular, no doctor is currently unmatched, so actually, DA will halt and return the matching μ'. □

Moreover, the following hints at a "lattice property" of stable matchings: in order for one side to benefit, the other side must be worst off.

Claim 1.11. *Let μ, μ' be stable matchings, and say $\mu(d) = h$, but $\mu'(d) \neq h$. Then $\mu'(d) >_d h$ if and only if $\mu'(h) <_h d$.*

Proof. ($\Leftarrow$) "If h downgrades, then d upgrades". Suppose $\mu'(h) <_h d$. Because μ' is stable, yet d and h are not matched in μ', we must have $\mu'(d) >_d h$, or else (d, h) would form a blocking pair. (A rephrasing: this direction is easy because the definition of stability immediately makes it impossible for d and h to both downgrade).

($\Rightarrow$) "If h upgrades, then d downgrades". Let $d' = \mu'(h) \neq d$ and $h' = \mu'(d) \neq h$. Suppose that $d' >_h d$, and for contradiction suppose that $h' >_d h$. Because μ' is stable, (d', h') is not a blocking pair, so either $h >_{d'} h'$ or $d >_{h'} d'$. In the first case, (d', h) form a blocking pair in μ, and in the second case, (d, h') form a blocking pair in μ. Thus, in either case μ is not stable.

□

1.1 Incentives

A surprisingly simple counterexample demonstrates that no procedure for assigning stable matchings can be dominant strategy incentive compatible for all agents:

m1	m2	m3	w1	w2	w3
w2	w1	w1	m1	m3	m1
w1	w2	w2	m3	m1	m2
w3	w3	w3	m2	m2	m3

x: { m1: w2, m2: w3, m3: w1 }
y: { m1: w1, m2: w3, m3: w2 }

w1	m2, m3	m3	m3
w2	m1	m1, m2	m1
w3			m2

m1	w1, w3	w1
m2		w3
m3	w2	w2

Above, x is the man-optimal and y is the woman-optimal stable outcome. However, if w1 reports the preference m1, m2, m3, then w1, m2 forms a blocking pair in x, while y remains stable. Indeed, the man-optimal outcome in this case is y, so y is the unique stable outcome. Thus, a stable matching procedure must return y, and w1 benefits from this deviation.

On the other hand, if m1 reports the preference w2, w3, w1, then m1, w3 form a blocking pair in y, while x remains stable. Indeed, the woman-optimal outcome in this case is x, so x is the unique stable outcome. Thus, a stable matching procedure must return x, and m1 benefits from this deviation.

Claim 1.12. *No stable matching mechanisms is DSIC for all agents.*

1.2 Linear Program Formulation

Let $A \subseteq D \times H$ be the set of "admissible pairs", i.e. those (d, h) for which both elements prefer being matched to each other to being unmatched.

$$\begin{array}{llll}
 & MP_u : & & \\
\text{variables:} & x_{dh} \geq 0 \qquad \forall d \in D, h \in H & & \\
\text{maximize:} & \displaystyle\sum_{i \in D, j \in H} x_{ij} & & \\
\text{subject to:} & \displaystyle\sum_{j \in H} x_{dj} \leq 1 & \forall d \in D & (\text{dual variable } \alpha_d) \\
 & \displaystyle\sum_{i \in D} x_{ih} \leq 1 & \forall h \in H & (\text{dual variable } \beta_h) \\
 & \displaystyle x_{dh} + \sum_{j >_d h} x_{dj} + \sum_{i >_h d} x_{ih} \geq 1 & \forall d \in D, h \in H & (\text{dual variable } \gamma_{dh})
\end{array}$$

$$\begin{array}{llll}
 & DMP : & & \\
\text{variables:} & \alpha_d \geq 0 \qquad \forall d \in D & & \\
 & \beta_h \geq 0 \qquad \forall h \in H & & \\
 & \gamma_{dh} \geq 0 \qquad \forall d \in D, h \in H & & \\
\text{minimize:} & \displaystyle\sum_{d \in D} \alpha_d + \sum_{h \in H} \beta_h - \sum_{d \in D, h \in H} \gamma_{dh} & & \\
\text{subject to:} & \displaystyle\alpha_d + \beta_h - \gamma_{dh} - \sum_{j <_d h} \gamma_{dj} - \sum_{i <_h d} \gamma_{ih} \geq 1 & \forall d \in D, h \in H & (\text{dual of variable } x_{dh})
\end{array}$$

The following lemmas give a major technical tool, and also distinguish this linear program from most others.

Lemma 1.13. *For any stable fractional matching x (i.e. a feasible point in MP) (α, β, x) is feasible for DMP, where*

$$\forall d \in D : \alpha_d = \sum_{j \in H} x_{dj} \qquad\qquad \forall h \in H : \beta_h = \sum_{i \in D} x_{ih}$$

Moreover, x has the same value in MP as (α, β, x) has in DMP, so every fractional matching x is an optimal solution of MP.

Proof. We clearly have $\alpha, \beta, x \geq 0$. Thus it suffices to check the dual constraint of each x_{dh}:

$$\sum_{j \in H} x_{dj} + \sum_{i \in D} x_{ih} - x_{dh} - \sum_{j <_d h} x_{dj} - \sum_{i <_h d} x_{ih} = x_{dh} + \sum_{j >_d h} x_{dj} + \sum_{i >_h d} x_{ih} \geq 1$$

Furthermore, the objective value of DMP works out to be:

$$\sum_{d \in D} \alpha_d + \sum_{h \in H} \beta_h - \sum_{d \in D, h \in H} x_{dh} = \sum_{d \in D} \sum_{j \in H} x_{dj} + \sum_{h \in H} \sum_{i \in D} x_{ih} - \sum_{d \in D, h \in H} x_{dh} = \sum_{d \in D, h \in H} x_{dh}$$

So by strong duality, x and (α, β, x) are both optimal. □

We now procede to reprove several of the classical results about stable matchings using this linear programming framework.

Claim 1.14. *I.e. the same set of agents is matched in every stable outcome.*

1.3 Many-to-one Matchings

Consider this counterexample from [Rot85], in which c1 has a quota of two students (other ci are taking one student). We run college-proposing DA:

c1	c2	c3
s1	s1	s3
s2	s2	s1
s3	s3	s2
s4	s4	s4

s1	s2	s3	s4
c3	c2	c1	c1
c1	c1	c3	c2
c2	c3	c2	c3

s1	c1 , c2	c1	c1	c1, c3	c3
s2	c1	c1, c2	c2	c2	c2
s3	c3	c3	c3, c1	c1	c1
s4					c1

The example was constructed so that c1 would "kick itself out" from s1 (by causing c3 to propose to s1). This implies two bad news for the strategic properties of many-to-one matchings: if c1 reports preference list s1, s2, s4, s3, then c1 is matched to s1, s4 (which it should strictly prefer to s3, s4). Thus, in many-to-one matching DA, the proposing side is not strategyproof (if the proposing side has quotas greater than 1).

For another bit of bad news is, consider the many-to-one matching c1: s2, s4; c2: s1, c3: s3. Every college prefers this matching to the result of college-proposing DA, contrary to claim 1.10. Note that indeed this matching is not stable: c1 would rather have s1 than s2, and s1 would rather have c1 than c2.

The following standard assumption is typically made about how preference over students relates to preferences over groups of students:

Definition 1.15. *A relation $>^*$ over sets of students is* responsive *if $\{s\} >^* \{t\}$ for $s, t \notin S$ implies $S \cup \{s\} >^* S \cup \{t\}$.*

Extensive properties are demonstrated in [RS89], I guess.

1.4 The Birkhoff Polytope

Definition 1.16. *A corner of a polytope $P \subseteq \mathbb{R}^n$ is a point $c \in P$ such that there does not exist $x, y \in P$, $x \neq c \neq y$, such that $c \in [x, y] = \{tx + (1-t)y : t \in [0,1]\}$*

Claim 1.17. *The diameter of the Birkhoff polytope graph is* 2

Proof. Every permutation is the product of two (non-disjoint) cycles:

$$\begin{aligned}&(i_1^1\, i_2^1 \dots i_{k_1}^1)(i_1^2\, i_2^2 \dots i_{k_2}^2)\dots(i_1^k\, i_2^k \dots i_{k_k}^k)\\ &= (i_1^1\, i_2^1 \dots i_{k_1}^1\; i_1^2\, i_2^2 \dots i_{k_2}^2\; i_1^3 \dots i_{k_{k-1}}^{k-1}\; i_1^k\, i_2^k \dots i_{k_k}^k)(i_1^k\, i_1^{k-1} \dots i_1^3\, i_1^2\, i_1^1)\end{aligned}$$

□

References

[Rot85] Alvin Roth. The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory*, 36(2):277–288, 1985.

[RS89] Alvin E. Roth and Marilda Sotomayor. The college admissions problem revisited. *Econometrica*, 57(3):559–570, 1989.