Deferred Acceptance with Compensation Chains

Piotr Dworczak[*]

November 9, 2017

Abstract

I introduce a class of algorithms called Deferred Acceptance with Compensation Chains (DACC). DACC algorithms generalize the DA algorithms of Gale and Shapley (1962) by allowing both sides of the market to make offers. The main result is a characterization of the set of stable matchings: a matching is stable if and only if it is the outcome of a DACC algorithm. DACC algorithms are an attractive alternative for matching markets in which the designer is concerned about fairness. The proof of convergence of DACC algorithms uses a novel technique based on a construction of a potential function.

Keywords: Stability, Deferred Acceptance, Potential Function, Fairness
JEL numbers: C78, D47.

---

[*]Becker Friedman Institute, University of Chicago, email: dworczak@uchicago.edu, website: http://home.uchicago.edu/∼dworczak. I would like to thank Nick Arnosti, Peter Biro, Yi Chen, Fuhito Kojima, Shengwu Li, Giorgio Martini, Paul Milgrom, Alexandru Nichifor, Afshin Nikzad, Michael Ostrovsky, Al Roth, Andrzej Skrzypacz, Bumin Yenmez, Anthony Lee Zhang, participants of the market design seminar at Stanford, EC'16 conference in Maastricht, and members of Paul Milgrom's research group for helpful comments and discussion.

1 Introduction

Deferred Acceptance (DA) algorithms play a central role in matching theory. In a seminal paper, Gale and Shapley (1962) used a men-proposing DA algorithm to show existence of a stable matching in the marriage problem. Stability has proven to be the key to designing successful matching markets in practice and is one reason why DA algorithms have gained so much prominence in market design.[1]

The Gale-Shapley algorithm produces the stable matching that is most preferred by agents on the proposing side. The men-proposing and the women-proposing versions achieve two extreme points of the set of stable outcomes. What happens when we allow for an arbitrary sequence of proposers? Is the outcome stable? Can all stable matchings be reached if we let both men and women propose? While these questions seem quite fundamental, the literature does not seem to provide final answers.[2]

I attempt to fill this gap by exploring the connection between stability and a general class of deferred acceptance algorithms (DACC). Just as in the Gale-Shapley DA, agents make and tentatively accept offers in a myopic way, taking into account only the currently available options. However, both sides of the market are allowed to make offers in an arbitrary order.

The paper makes three distinct contributions. First, I show that there exists a natural generalization of the Gale-Shapley algorithm, namely the class of DACC algorithms, with the property that the outcome of each algorithm in the class is stable, and every stable outcome can be reached by an algorithm from the class. Conceptually, this result can be seen as establishing an equivalence between (properly generalized) deferred acceptance procedures and stability. Second, on the practical side, I argue that the DACC class can be an attractive matching algorithm for markets in which the designer is concerned about fairness. Because DACC allows for an arbitrary sequence of proposers, there exist DACC algorithms that treat the two sides of the market symmetrically. This "procedural fairness" is complementary to other notions of fairness, such as the median matching which can be viewed as a "fairness of outcomes." Importantly for applications, the DACC algorithm extends to many-to-one matching with

[1] See Roth (2007). Extensions of the deferred acceptance algorithm are used in public high-school choice in New York (Abdulkadiroğlu, Pathak and Roth, 2005a) and Boston (Abdulkadiroğlu, Pathak, Roth and Sönmez, 2005b), allocating medical students to residencies (NRMP) as well as in other medical labor markets surveyed by Roth (2007). A stable mechanism is advocated for cadet-branch matching in the US army by Sönmez and Switzer (2013).

[2] I review papers giving partial answers to the above questions in Section 5.

contracts. Third, the methodological contribution of the paper is to provide a novel proof technique of convergence. To generate all stable matchings, the DA procedure can no longer feature a monotone offer process, so the standard argument for convergence does not apply. I construct a potential function and show its monotonicity along the paths of the algorithm. I conjecture that similar constructions can be useful in analyzing convergence of other matching systems, especially in settings when there is not enough structure in the offer process (such as decentralized markets).

A Deferred Acceptance with Compensation Chains (DACC) algorithm works roughly as follows. Agents make offers one at a time according to a pre-specified (arbitrary) order. Agents propose to the best available partner, and hold an offer if they prefer the proposer to the current match. Partners become unavailable to agent i when they reject or divorce i, and become available when they propose to i. Initially, all agents on the other side of the market are available to any agent i. The algorithm stops when everyone is matched to the best available partner.

When both men and women propose, it is possible that a woman rejects an offer from a man but proposes to him later on. As a consequence, the man might "withdraw" an offer he made to another woman, an event I call deception. This non-monotonicity of the offer process may upset convergence and stability, but the compensation chains that I introduce restore them. Compensation consists in allowing the deceived agent to make an additional offer (out of turn).

Convergence is established by constructing a potential function. For any agent i, count the number of slots between the current match and the best available partner according to agent i's preference ranking. The potential function is the sum of these numbers across all agents. The algorithm stops when the potential function hits zero, and the matching is stable in this case. I show that the function is decreasing (strictly in some steps) after sufficiently many rounds. Intuitively, the potential function goes down for agents whose offers are accepted or rejected. However, it may go up for agents that are divorced. After sufficiently many rounds, every divorce is a deception and hence triggers a compensation chain. When the chain stops, the potential function falls below its original level so its monotonicity is preserved.

I define the DACC algorithm and study its properties in Section 3. DACC reduces to the Gale-Shapley algorithm if only one side makes offers. It converges in finite time, and the outcome is stable. Every stable matching can be achieved by choosing an appropriate order over agents.

In Section 4, I characterize the DACC algorithm as a special case of a procedure in which non-monotone operators are applied recursively. In a general setting, I provide a necessary and sufficient condition under which decentralized procedures (individual agents taking actions sequentially) converge to a stable outcome of the aggregate system. I discuss the connection to papers that characterize stable matchings as fixed points. While those papers typically rely on monotone operators and Tarski's fixed-point theorem, convergence of DACC is similar in spirit to stability of the tâtonnement process for prices in competitive equilibrium.

Because DACC allows for an arbitrary order over proposers, it is fairly flexible. Section 5 argues that some well-known algorithms can be interpreted as special cases of the DACC algorithm with a particular order in which agents apply. I also discuss additional properties of DACC and compare it to other algorithms from the matching literature.

Section 6 explains why DACC needs to be sufficiently complicated to have the desirable properties mentioned above. Simpler generalizations of the Gale-Shapley algorithm without compensation chains are introduced, and I discuss why they fail to achieve all of the properties of DACC.

The flexibility of DACC is explored further in Online Appendices C and D. In Appendix C, I show that DACC can be used without any modifications in the roommates problem. Any stable matching can be an outcome of DACC, and under a standard sufficient condition for existence of a stable matching, DACC converges to a stable outcome. In Appendix D, I extend DACC to many-to-one matching with contracts. If contracts are substitutes in the sense of Hatfield and Milgrom (2005), DACC is guaranteed to converge to a stable outcome. The construction and proof provide new insights about the role of the substitutes condition. Rather than guaranteeing monotonicity of an aggregate operator, substitutability of preferences allows for a decentralized procedure in which one contract at a time is proposed.[3]

[3] This observation is related to Gul and Stacchetti (1999) who provide a connection between substitutability and a single-improvement property for Walrasian equilibrium with indivisible objects. Under a stronger assumption (satisfied by responsive preferences in many-to-one matching), I show that all stable allocations can be achieved by DACC.

2 Preliminaries

There is a finite set of men M and a finite set of women W. N is the set of all agents, and for $i \in N$, I let N_i denote W if $i \in M$, and M if $i \in W$. Each agent $i \in N$ is endowed with a preference relation $\succ^i$ on $N_i \cup \{\emptyset\}$, where $\emptyset$ represents the outside option of being unmatched. For ease of exposition, I assume that preferences are strict. A matching μ is a set of unordered pairs $\{i, j\}$ such that if $i \in N$, then $j \in N_i \cup \{\emptyset\}$ and each agent $i \in N$ appears in exactly one pair. With slight abuse of notation, I write $\mu(i) = j$ when $\{i, j\} \in \mu$. I will say that agent i is matched when $\mu(i) \in N_i$, and that i is unmatched if $\mu(i) = \emptyset$.

Agent $j \in N_i$ is acceptable to i if $j \succ^i \emptyset$. A matching μ is stable if all agents are matched to acceptable partners or remain unmatched, and $j \succ^i \mu(i)$ implies $\mu(j) \succ^j i$, for all $i \in N$, $j \in N_i$.

A budget set B_i for agent i is any subset of N_i and the outside option $\emptyset$ (which is always available to agents). The budget system $\mathcal{B} = \{B_i\}_{i \in N}$ is said to support a matching μ if, for every agent i, $\mu(i) = \operatorname{argmax} \{B_i; \succ^i\}$.[4] The connection between a budget system and stability is captured by the following observation.

Observation 1. *Suppose that the budget system $\mathcal{B} = \{B_i\}_{i \in N}$ supports a matching μ. If*

$$\{j \in N_i : i \succeq^j \mu(j)\} \subset B_i \tag{2.1}$$

holds for all $i \in M$ or for all $i \in W$, then μ is stable.

Condition (2.1) says that the budget set of agent i contains all agents who weakly prefer i to their $\mu-$partner.

I conclude this section with two remarks about the Gale-Shapley algorithm which I will sometimes refer to as One-Sided Deferred Acceptance (1DA). First, the order in which men propose in 1DA does not play any role. Instead of simultaneous proposals, men could apply one-by-one, and women could make (tentative) acceptance decisions by evaluating the proposer against their current match.[5] Second, 1DA can be described in the language of budget sets.[6] In the men-proposing version, each men starts with a budget set consisting of all women, and each woman starts with an empty budget

[4] Here, $\operatorname{argmax} \{A; \succ\}$ is defined as the the most preferred option from the set A with respect to the order $\succ$.

[5] See McVitie and Wilson (1971) for an algorithm based on this observation.

[6] This fact is well known, and has been exploited for example in Hatfield and Milgrom (2005).

set. In every round, each man applies to the most preferred woman in his budget set. Applicants to a woman are added to her budget set and she chooses the most-preferred partner from her budget. If a man is rejected by a woman, she is discarded from that man's budget set. The final matching is stable because equation (2.1) holds for all women.

3 Deferred Acceptance with Compensation Chains (DACC)

DACC generalizes 1DA by allowing both men and women to make offers in some pre-specified order. Formally, fix a sequence $\Phi : \mathbb{N} \to N$ such that each value in N is taken infinitely many times. Whenever I refer to a sequence Φ in this paper, I assume that Φ has this property. DACC(Φ) is defined in frame Algorithm 1. An informal description is given below. I will often omit the argument Φ and refer to "the DACC algorithm" assuming implicitly that Φ has been fixed.

Every agent starts with a full budget set $B_i = N_i$, and the initial matching μ is empty. The budget system $\{B_i\}_{i\in N}$ and the matching μ are adjusted during the course of the algorithm. I say that "i is divorced by j" (or "j divorces i") when i and j were matched and then j broke the match with i in order to be matched to a more preferred partner (i became unmatched).

Proposals and Acceptance. In round k, agent $i = \Phi(k)$ makes an offer to the most preferred person j in his or her budget set.[7] Agent j (tentatively) accepts if i is preferred to j's current match (or to the outside option if j is unmatched). In that case, we adjust μ by matching i and j, and divorcing their previous partners (if they had any). Otherwise, j rejects i and the matching μ is unchanged.

Budget Sets. Whenever i proposes to j, we add i to j's budget set B_j. Whenever i is rejected or divorced by j, we remove j from i's budget set B_i.

Compensation Chains (CCs). I say that i *deceived* j if i divorced j to whom i has proposed before. Whenever some i deceives j, we *compensate* agent j. That is, j is allowed to make an offer in the current round irrespective of the order Φ. If j is accepted by k who by doing so deceives $\mu(k)$ (k's current match), then $\mu(k)$ is compensated, i.e. allowed to propose next. This chain of compensations ends when the last person in the chain exhausts his or her budget set, or is accepted by agent l who does not deceive $\mu(l)$ (for example when $\mu(l) = \emptyset$). Then, the algorithm proceeds to the next round,

[7] If i is already matched to j, or if there are no acceptable partners in i's budget set, we skip the round.

and the proposer is determined by Φ. Formally, to identify "deceptions", I keep track of a set A_i, for each i, which is initially empty, and records all agents who propose to i as the algorithm progresses.

The algorithm stops when all agents are matched under μ to the best option in their current budget set. If a stable matching is reached, all subsequent proposals are rejected but formally the algorithm continues until the above stopping criterion is satisfied.

The definition is illustrated with examples in Appendix B. The reader may find it helpful to compare DACC to its simpler but inferior (in terms of properties) versions defined in Section 6.

The first two parts of the description directly generalize 1DA to a two-sided deferred acceptance procedure. To understand the addition of compensation chains, note that in 1DA any offer is effectively available to the receiver till the end of the algorithm. An offer to a woman in a men-proposing 1DA immediately becomes a lower bound on her final match utility. This monotonicity drives the convergence of 1DA to a stable outcome. With two-sided offers, we cannot guarantee that property. A proposer may withdraw an offer if he or she later receives an offer from a preferred partner, an event that I called "deception". CCs are a way to partially restore monotonicity by compensating agents for the loss of a withdrawn offer.

Because deceptions never take place if only one side of the market applies (and hence there are no CCs), DACC generalizes 1DA. Formally, if only men appear in Φ initially for sufficiently many rounds, the algorithm is effectively identical to the men-proposing 1DA, and it converges to the men-optimal stable outcome.

The main result of the paper is Theorem 1.

Theorem 1. *For any sequence Φ, DACC(Φ) stops in finite time and its outcome μ is stable. Conversely, for an arbitrary stable matching μ, there exists a sequence Φ such that μ is the outcome of DACC(Φ). Therefore, a matching is stable if and only if it is the outcome of a DACC algorithm.*

The remainder of this section proves Theorem 1 in a series of claims.

Claim 1. *If DACC stops, the outcome is stable.*

Proof. Suppose not. Then there is a blocking pair $\{i, j\}$. By the stopping criterion, there exists the last time τ in the algorithm when i and j interacted. That is, up to

Algorithm 1 Deferred Acceptance with Compensation Chains - DACC(Φ)

MAIN BLOCK

1. **For** $i \in N$ **set** $B_i := N_i$ **and** $A_i := \emptyset$; (B_i - budget set of i; A_i - agents who applied to i)
2. **Set** $\mu := \emptyset$ **and** $CC := \emptyset$; (CC keeps track of agents that need to be compensated[a])
3. **Set** $k := 1$ **and** $t := 1$; (k keeps track of rounds and t keeps track of time)
4. **While** $\exists i \in N$, $\mu(i) \prec^i \text{argmax}\{B_i; \succ^i\}$ **do:**
   (a) **If** $CC = \emptyset$ **then:** (If there are no agent to be compensated)
      i. $i := \Phi(k)$;
      ii. i applies;
      iii. $k := k + 1$; (Update the round number)
   (b) **else:**
      i. $i :=$ take from the top of CC (Compensate the agent at the top of the stack)
      ii. i applies;
      iii. **If** $\mu(i) \neq \emptyset$ **or** $B_i = \emptyset$ **then** remove i from CC;
   (c) $t := t + 1$. (Update physical time)

[a] CC has a stack structure; the agent at the top of CC is next to be compensated.

Description of the procedure "i applies"

1. **Set** $j := \text{argmax}\{B_i; \succ^i\}$; ($i$ applies to j)
2. **If** $\{i, j\} \in \mu$ **or** $j = \emptyset$ **then** return; **else:** (If i and j are already matched or i applies to $\emptyset$)
3. **Set** $A_j := A_j \cup \{i\}$ **and** $B_j := B_j \cup \{i\}$; (Record that i applied to j and increase j's budget)
4. **If** $i \succ^j \mu(j)$ **then:**[a] (If i is accepted by j)
   (a) **If** $\exists j' \neq j$ such that $\{i, j'\} \in \mu$ **then:** (If i was matched to someone)
      i. **If** $i \in A_{j'}$ **then** add j' to the top of CC**;** (Compensate j' if i deceives j')
      ii. $B_{j'} := B_{j'} \setminus \{i\}$;
      iii. $\mu := \mu \setminus \{i, j'\}$; (divorce i and j')
   (b) **If** $\exists i' \neq i$ such that $\{j, i'\} \in \mu$ **then:** (If j was matched to someone)
      i. **If** $j \in A_{i'}$ **then** add i' to the top of CC**;** (Compensate i' if j deceives i')
      ii. $B_{i'} := B_{i'} \setminus \{j\}$**;**
      iii. $\mu := \mu \setminus \{j, i'\}$; (divorce j and i')
   (c) $\mu := \mu \cup \{\{i, j\}\}$; (match i and j)
5. **else:** $B_i := B_i \setminus \{j\}$. (If i is rejected by j, remove j from i's budget)

[a] Items (a) and (b) can be executed in any order, even random, i.e. if there are two CCs, it does not matter which one is run first.

relabeling, either *(i)* i applied to j and was rejected, or *(ii)* i and j were matched and j divorced i. In both cases, $i \in B_j$ after τ, and hence also when the algorithm stops. Indeed, in case *(i)* i is added to j's budget set because i applies to j, and in case *(ii)* this follows from the fact that whenever agents are matched, they have each other in their respective budget sets. But $i \in B_j$ is a contradiction with the stopping criterion. Because $\{i, j\}$ is a blocking pair, j must be matched to someone less preferred to i, despite i being in j's budget set. □

The proof is a direct generalization of the argument used by Gale and Shapley (1962), expressed in the language of budget sets. A careful inspection shows that $i \in B_j$ or $j \in B_i$, for all $i \in N$, $j \in N_i$, at all times in DACC. If $i \succ^j \mu(j)$ when the algorithm stops, then $i \notin B_j$, so $j \in B_i$. Thus, equation (2.1) holds for all agents once DACC terminates.

To state the next claim, I have to make precise what I mean by a (single) CC. Consider an instance in the k-th round of the DACC algorithm when $\Phi(k)$ applies and causes a divorce of some agent i by $j \in A_i$ (i.e. j deceives i).[8] Then we initiate a CC at i. Let $i_0 = i$. Fixing a sequence of agents $(i_0, i_1, ..., i_{m-1})$ who applied in that CC so far, I show how to choose i_m. If i_{m-1} applied and was rejected, choose $i_m = i_{m-1}$. If i_{m-1} applied and was accepted by j who deceived agent l, set $i_m = l$ (now l will be compensated). In all other cases, terminate the CC.

Claim 2. *Every CC stops in finite time.*

Proof. The claim follows from two observations. First, in a CC initiated at a man, only men propose (analogously for women). Second, in a CC where men propose, budget sets of men never grow, and in every round of the CC in which it doesn't terminate, a budget set of some man shrinks. If the CC does not terminate for other reasons, it terminates because the budget set of some man proposing in the CC becomes empty.□

Claim 3. *The DACC algorithm stops in finite time.*

Proof. Fixing Φ, let $(\mathcal{B}^k, \mu^k)$ be the budget system and matching at the end of round k in the DACC(Φ) algorithm. I introduce the following function for each agent $i \in N$:

$$d_i(\mathcal{B}^k, \mu^k) = |\{j \in B_i^k : j \succ^i \mu^k(i)\}|. \tag{3.1}$$

[8] It could be either that $\Phi(k) = j$, i.e. i and $\Phi(k)$ were matched, or that $\Phi(k)$ applied to j who was matched to i. In every round, we can have zero, one, or two CCs.

The function d_i counts the agents in i's budget set that i prefers to his or her current match. Because no agent is ever matched to a partner who is not in the budget set, the stopping criterion is satisfied if and only if

$$d(\mathcal{B}^k,\, \mu^k) := \sum_{i \in N} d_i(\mathcal{B}^k,\, \mu^k) = 0. \tag{3.2}$$

In light of Claim 1, the function d measures the distance to stability.

Lemma 1. *Fixing Φ, there exists a strictly increasing sequence of positive integers $(a_k)_{k\in\mathbb{N}}$ such that d is strictly decreasing along the sequence $(\mathcal{B}^{a_k},\, \mu^{a_k})_{k=1,2,\ldots}$, for all k such that DACC(Φ) hasn't yet stopped in round a_k.*

The proof of the Lemma is technical and hence relegated to Appendix A. I sketch it below. By direct inspection, the function d_i decreases weakly when agent i receives an offer, and decreases strictly when agent i applies. Thus, d declines in every round of the algorithm in which there are no divorces. I show that after sufficiently many periods, every divorce leads to a CC. This rules out a loop involving non-deceptive divorces. When a CC stops, all agents who applied in the CC are matched to the most preferred option in their budget set, i.e. d_i attains value 0 for such agents. In particular, d_i must have gone weakly down. Hence, d is strictly decreasing along $(\mathcal{B}^{a_k},\, \mu^{a_k})_{k=1,2,\ldots}$, where the restriction to a subsequence a eliminates rounds k when the stopping criterion is already satisfied for $\Phi(k)$ (i.e. $d_{\Phi(k)} = 0$).

By Lemma 1, the distance to stability declines as the algorithm progresses. Because the function d is bounded above by $2 \cdot |W| \cdot |M|$, there must exist a finite time K such that $d(\mathcal{B}^K,\, \mu^K) = 0$. Thus, the algorithm stops at K. □

Remark 1. It is clear from the proof that there is some flexibility in specifying when CCs should be run. For example, if *(i)* we run a CC after *every* divorce, or *(ii)* we run CCs only after some round $k^\star$ (where $k^\star$ could be random, endogenous etc.), then DACC will still converge to a stable matching in finite time.

The following observation will be relevant for the discussion in Section 5. It follows directly from the proofs of Claims 1-3 which made no use of the fact that the initial matching is empty.

Observation 2. *If DACC starts at an arbitrary matching, and initial budget sets satisfy $i \in B_j$ or $j \in B_i$ for all $i \in N$, $j \in N_i$, then the algorithm will converge in finite time to a stable matching.*

The final claim establishes the converse part of Theorem 1.

Claim 4. *For any stable μ, there is an ordering Φ such that μ is the outcome of DACC(Φ). Moreover, μ can be achieved with an order Φ that does not lead to any compensation chains.*

Proof. Fix μ that is stable. Say that $j \in N_i$ is the $\mu-$partner of i if $\{i, j\} \in \mu$. I construct Φ recursively. Choose $\Phi(1)$ to be an arbitrary agent $i \in N$. In round $k+1$, if the DACC algorithm hasn't stopped, I choose $\Phi(k+1)$ as a function of what happened when $\Phi(k)$ applied in round k:

1. if $\Phi(k)$ was rejected, set $\Phi(k+1) = \Phi(k)$;
2. if $\Phi(k)$ was accepted by his or her $\mu-$partner, set $\Phi(k+1)$ to be an arbitrary agent who is not currently matched to the $\mu-$ partner;
3. if $\Phi(k)$ was accepted by j who is not his or her $\mu-$partner, set $\Phi(k+1) = j$.

I prove that in any round k, the following properties hold:

(a) The set of matches at the end of round k consists of pairs in μ and at most one pair that is not in μ. If such pair exists, it involves the agent $\Phi(k+1)$ who proposes next.

(b) Up to (and including) round k, there haven't been any CCs.

(c) Up to (and including) round k, $\mu(i) \in B_i$, $\forall i \in N$ (no agent was rejected by their $\mu-$partner).

If the above properties hold for all k until the DACC algorithm stops at K, then we are done. Because property (c) holds at K, it cannot be that some agents who are matched under μ remain unmatched (that would contradict the stopping criterion). By property (a), there can exist at most one pair that is not in μ. If it did, then by property (c) and the stopping criterion we would get a contradiction with stability of μ (agents in that pair would prefer each other to their respective $\mu-$partners).

I prove properties (a)-(c) by induction over k. For $k = 0$ (before the algorithm starts) the claim is obvious. Suppose that the claim holds up to and including round k. Consider round $k+1$.

Let $i = \Phi(k+1)$ and suppose a stable matching hasn't been reached yet. By the choice of Φ and the inductive hypothesis (property (a)), i is not matched to his or her $\mu-$ partner. Moreover, once i divorces the current partner (assuming i has one), all matched pairs will be in μ. Agent i applies. If i is rejected, properties (a)-(c) are obviously satisfied (i cannot be rejected by $\mu(i)$ because $\mu(i)$ is not matched). If i is accepted, property (a) follows from the inductive hypothesis and the way we choose Φ, property (c) is obvious, and property (b) follows from two observations. First, by the inductive hypothesis (property (c)), i never applied to someone less preferred to $\mu(i)$. In particular, in round $k+1$, i applies to some j that i prefers weakly to $\mu(i)$. By stability of μ, i cannot be accepted by any matched agent (as all matched agents are matched to their $\mu-$partners), so j was unmatched. Thus j did not divorce anybody. Second, if i was matched to some agent l, it must be that l applied to i in round k. Thus, by property (c), l prefers i to $\mu(l)$. If i applied to l before, we would have that i prefers l to $\mu(i)$ which contradicts stability of μ. Hence $i \notin A_l$. It follows that this divorce could not lead to a CC. $\square$

4 A Fixed-Point Approach

It is well known (see Adachi, 2000, Fleiner, 2003, Hatfield and Milgrom, 2005, Echenique and Oviedo, 2006) that the set of stable matchings can be closely related to the set of fixed points of a monotone operator.[9] Theorem 1 characterizes stability as a set of outcomes that arise when agents apply one-by-one according to a decentralized procedure. I aim to draw a connection between these two approaches. I introduce an abstract setting, and then show that DACC is a special case of a more general procedure.

Take an arbitrary finite set $\mathcal{X}$, and suppose we are interested in finding a fixed point of an operator $\Psi : \mathcal{X} \to \mathcal{X}$. Let $X^\star = \{x \in \mathcal{X} : \Psi(x) = x\}$. Assume that there exist operators $(\Psi_i)_{i \in N}$ on $\mathcal{X}$ such that

$$(\Psi(x) = x) \iff (\Psi_i(x) = x,\ \forall i \in N)\,, \forall x \in \mathcal{X}. \tag{4.1}$$

I will say that $(\Psi_i)_{i \in N}$ decentralize Ψ. In a typical application, Ψ is an operator describing aggregate behavior of some economic system, and Ψ_i is an action taken by

[9] Recently, a number of papers, for example Azevedo and Hatfield (2015) and Che, Kim and Kojima (2015), use topological fixed-point theorems to analyze stability in large matching markets. Their constructions do not rely on monotonicity but instead exploit the tractability of continuous models.

agent i in that system (N is the set of agents). I would like to see whether a decentralized procedure in which agents take actions sequentially (i.e. applying operators Ψ_i recursively) can lead to a stable outcome of the system (i.e. a fixed point of Ψ).

Let Λ be a subset of $N^{\mathbb{N}}$. A generic element $\phi \in \Lambda$ specifies the order in which agents take actions. I say that $f : \mathcal{X} \to \mathbb{N}$ is a potential function[10] for the system $((\Psi_i)_{i\in N}, \Lambda, x_0)$ if *(i)* $f(x) = 0 \iff x \in X^\star$, and *(ii)* for every $\phi \in \Lambda$, there exists an increasing sequence of positive integers $(a_k)_{k\in\mathbb{N}}$ such that f is strictly decreasing along the sequence $(x_{a_k})_{k=1,2,\ldots}$, for all k such that $f(x_{a_k}) > 0$, where x_k is defined recursively by

$$x_k = \Psi_{\phi(k)}(x_{k-1}), \tag{4.2}$$

for all $k \geq 1$. Existence of a potential function rules out cycles in the procedure.

Observation 3. *For a fixed $x_0 \in \mathcal{X}$, the sequence (x_k) defined by (4.2) converges in finite time to a point in $X^\star$ for any $\phi \in \Lambda$ if and only if the system $((\Psi_i)_{i\in N}, \Lambda, x_0)$ admits a potential function.*

Proof. Sufficiency follows from the argument used at the end of the proof of Claim 3. Necessity is trivial because we can take $f = \mathbf{1}_{\{x \notin X^\star\}}$ and sequence a with $a_1 = \min\{k \in \mathbb{N} : x_k \in X^\star\}$, where (x_k) is defined by equation (4.2) (the rest of the sequence a can be chosen arbitrarily). $\square$

To see how this abstract result relates to the matching model, define $\mathcal{X}$ as

$$\mathcal{X} = \{(\mathcal{B}, \mu) : \mathcal{B} \text{ is a budget system, } \mu \text{ is a matching, } i \in B_j \text{ or } j \in B_i,\ \forall i \in N,\ j \in N_i\}$$

The starting point x_0 is $(\mathcal{B}_0, \emptyset)$, where $\mathcal{B}_0$ specifies a full budget for every agent. I define $\Psi_i : \mathcal{X} \to \mathcal{X}$, in the following way. $\Psi_i(\mathcal{B}, \mu)$ is the outcome which arises in the DACC algorithm when agent i applies to the most preferred option in B_i when the current matching is μ (for a formal definition, see the description of the procedure "i applies" in frame Algorithm 1).[11]

Proposition 1. *If $\Psi_i(\mathcal{B}^\star, \mu^\star) = (\mathcal{B}^\star, \mu^\star)$, for all $i \in N$, then $\mathcal{B}^\star$ supports $\mu^\star$ and $\mu^\star$ is stable. Conversely, if $\mu^\star$ is stable, there exists a budget system $\mathcal{B}^\star$ that supports $\mu^\star$ and such that $\Psi_i(\mathcal{B}^\star, \mu^\star) = (\mathcal{B}^\star, \mu^\star)$, for all $i \in N$.*

[10] In the theory of differential equations, such a function is often called the Lyapunov function.

[11] This procedure preserves the property that $i \in B_j$ or $j \in B_i$, $\forall\, i \in N$, $j \in N_i$, so Ψ_i is indeed a mapping from $\mathcal{X}$ into itself. Note that any point in $\mathcal{X}$ could be taken as the starting point x_0, consistent with Observation 2.

Proof. I first show that if $\Psi_i(\mathcal{B}^\star, \mu^\star) = (\mathcal{B}^\star, \mu^\star)$, for all $i \in N$, then $\mathcal{B}^\star$ supports $\mu^\star$. Suppose not. Then, either *(i)* $\mu^\star(j) \succ^j \operatorname{argmax}\{B^\star_j; \succ^j\}$, or *(ii)* $\mu^\star(j) \prec^j \operatorname{argmax}\{B^\star_j; \succ^j\}$, for some $j \in N$. In both cases we get a contradiction with $\Psi_j(\mathcal{B}^\star, \mu^\star) = (\mathcal{B}^\star, \mu^\star)$ because according to Ψ_j, j makes an offer to $\operatorname{argmax}\{B^\star_j; \succ^j\}$, and so either $B^\star_j$ or $\mu^\star(j)$ has to change. Once we know that $\mathcal{B}^\star$ supports $\mu^\star$, stability of $\mu^\star$ follows immediately from the discussion directly following the proof of Claim 1 and from Observation 1. For the converse part, fix a stable $\mu^\star$, and define the budget set of agent i, for each $i \in N$, by

$$B^\star_i = \{j \in N_i : i \succeq^j \mu^\star(j) \text{ or } \mu^\star(i) \succ^i j\}.$$

By stability of $\mu^\star$, $\mathcal{B}^\star$ supports $\mu^\star$, and $\Psi_i(\mathcal{B}^\star, \mu^\star) = (\mathcal{B}^\star, \mu^\star)$, for all $i \in N$. It is easy to check that $(\mathcal{B}^\star, \mu^\star) \in \mathcal{X}$. $\square$

CCs are effectively a modification of the order in which agents propose. We can obtain the set Λ for the matching problem by mapping each original Φ into the actual ex-post (i.e. including CCs) order in which agents proposed in DACC(Φ). The DACC algorithm can then be interpreted as iterative application of operators Ψ_i in the modified order. By restricting orders Φ to lie in Λ, I was able to find a potential function d defined by (3.1) and (3.2) for the system $((\Psi_i)_{i\in N}, \Lambda, x_0)$.

Hatfield and Milgrom (2005) provide a tight connection between the set of stable matchings and the set of fixed points of a monotone operator in a more general setting of many-to-one matching with contracts. If only one side of the market applies in DACC, the outcome of the algorithm and the operators Ψ_i are equivalent to what Hatfield and Milgrom (2005) call the *Generalized Gale-Shapley Algorithm.*[12] However, they do not define an iterative procedure that achieves other, non-extreme stable matchings. In fact, this cannot be done by applying a monotone operator on a lattice starting from an extreme element, because Tarski's fixed point theorem predicts that such a procedure converges to an extreme fixed point. That is why I assumed no such structure. The non-monotone operators Ψ_i can be seen as a decentralization of the Hatfield-Milgrom operator in the sense defined by relation (4.1). Note that DACC produces a matching in every step of the algorithm unlike some other algorithms with pre-matchings (see, for example, Adachi, 2000).

Convergence of DACC is reminiscent of the tâtonnement process for prices in com-

[12] See Appendix D for an extension of DACC to many-to-one matching with contracts.

petitive equilibrium. Arrow and Hurwicz (1958) and Uzawa (1960) show that, under conditions, a competitive equilibrium can be found by sequentially adjusting prices of individual goods in the economy. Their potential functions are defined on trajectories of differential equations describing the movement of prices, while this paper provides a discrete analog that is defined on "trajectories" of a matching algorithm.[13]

5 Generalizations and Comparison to other Algorithms

Knuth (1976) showed that if one starts from an unstable matching and satisfies blocking pairs in an arbitrary order, a stable outcome need not be reached. Roth and Vande Vate (1990) demonstrated that if blocking pairs are satisfied in random order, then the procedure converges with probability one to a stable outcome. They prove it by constructing a sequence of blocking pairs which, if satisfied, lead to a stable outcome.[14] An alternative proof of this result is provided by the DACC algorithm. It is enough to run the DACC algorithm starting at an arbitrary matching for any sequence Φ. By Observation 2, the outcome is stable. Ma (1996) shows that a random order mechanism based on Roth and Vande Vate (1990) does not achieve all stable matchings. By Claim 4, if we start from an empty matching and satisfy blocking pairs according to the DACC procedure, we can reach all stable outcomes.

DACC is a natural extension of the Gale-Shapley algorithm in that it preserves the logic of applications and rejections, is easy to describe, and does not rely on structural elements or auxiliary mathematical constructions. To the best of my knowledge, it is the first such class of algorithms with the feature that the outcome of each algorithm is stable, and each stable outcome can be achieved by an algorithm from the class. There are papers that characterize the set of stable matchings using non-deferred-acceptance algorithms relying on more complex mathematical objects: Irving and Leather (1986) use rotations, Adachi (2000) and Hatfield and Milgrom (2005) - pre-matchings and fixed-point theory on lattices, Kuvalekar (2015) - graph theory.

An important extension of the Gale-Shapley algorithm has been provided by McVitie and Wilson (1971). In fact, McVitie-Wilson propose two procedures. The first McVitie-Wilson algorithm can be seen as a sequential version of the (one-sided) Gale-

[13] For more recent examples of using potential functions to show convergence in auction settings, see Ausubel (2006) and Sun and Yang (2009).

[14] Kojima and Ünver (2008) extend the result of Roth and Vande Vate (1990) to many-to-many matching under suitable assumptions on preferences. Their construction of the converging sequence of blocking pairs uses ideas similar to the notion of compensation in DACC.

Shapley algorithm in which, using the language of the current paper, every rejection and divorce triggers a compensation chain. I showed that far less frequent compensations (only following deceptions) are sufficient to obtain stability, even if both sides of the market propose. Due to an arbitrary choice of proposers in most rounds, DACC generates all stable matchings, unlike the McVitie-Wilson algorithm. The second McVitie-Wilson algorithm[15] does find the set of all stable matchings by, roughly speaking, repeatedly running the first version of the algorithm with properly truncated preference rankings for the side of the market receiving offers. However, this second procedure produces non-stable matchings that have to be "manually" discarded. In contrast, every DACC algorithm is guaranteed to produce a stable matching in every instance of the problem.

The McVitie-Wilson procedure is superior to DACC for numerical computation of the set of all stable matchings. While DACC can be shown to run in polynomial time for any fixed order, running DACC repeatedly for all possible orders is not a computationally efficient procedure for generating all stable matchings. Irving and Leather (1986) provide results on efficient computation of the full set of stable matchings in the marriage model. By extending the McVitie-Wilson algorithm, Martínez *et al.* (2004) find all stable outcomes in a many-to-many matching model.

A related algorithm, called the "compromise algorithm", is proposed in an unpublished manuscript by Kesten (2004) for the setting of one-to-one matching in a balanced market. In the compromise algorithm, agents apply on-by-one, and the resulting matching is always stable. The order over proposers is determined by assigning ranks to agents and giving priority in applying to higher-ranked agents (over lower-ranked agents) every time they become unmatched. Using the language of DACC, following a divorce, the compromise mechanism compensates higher-ranked agents rather than deceived agents. This rule gives the compromise mechanism the monotonicity structure possessed by the original Gale-Shapley algorithm. For the same reason, the compromise algorithm may not generate all stable matchings.

An immediate extension of the DACC algorithm can be obtained by restricting the set of agents available in any round. This allows to capture the possibility that agents arrive gradually to the market. Formally, let $(P_k)_k$ be a non-decreasing sequence of subsets of N with the property that $P_k = N$ for sufficiently large k (i.e. all agents

[15] McVitie and Wilson (1971) give two versions of this procedure but the distinction between them is irrelevant for the discussion below. An algorithm with very similar properties, based on the idea of backtracking, has been proposed by Wirth (1978).

eventually arrive to the market). Only agents in P_k are present in the market in round k. To run DACC, we define budget sets in round k as $B_i^k \cap P_k$. It is clear from the proofs that we will reach a stable matching in finite time.

Blum and Rothblum (2002) (see also Blum, Roth and Rothblum, 1997) consider a model in which agents arrive to a stable matching market sequentially. The arriving agent applies to the most preferred partner with whom she forms a blocking pair (if such partner exists). If some agent is divorced as a result, she will now be matched to the most preferred partner with whom she forms a blocking pair. Blum and Rothblum (2002) call this process *the greedy correcting procedure* and show that it converges and restores stability in the larger market. The greedy correcting procedure can be seen as a special case of a CC, in the sense that the two would be observationally equivalent in the sequential arrival model of Blum and Rothblum (2002), assuming that we run the CC after every divorce (see Remark 1). The extension of DACC considered above allows us to interpret the Sequential Greedy Correcting procedure of Blum and Rothblum (2002) as a realization of a DACC algorithm for a particular sequence Φ, where we choose the sets P_k to reflect the arrival of agents to the market.

Blum and Rothblum (2002) show that agents prefer to arrive late in their model. In particular, the last arriving agent obtains the most preferred stable match partner. This implies that, unlike DACC, their algorithm cannot reach all stable matchings. On the other hand, the class of orders allowed by DACC is too large to obtain similar comparative statics result. It is easy to construct counterexamples to claims that in DACC agents prefer to come earlier or later in the sequence Φ.

The DACC algorithm is not strategy-proof for either side of the market in general. Note that strategy-proofness of a stable algorithm depends solely on which stable matching it eventually selects. The results of Sönmez (1999) can be used to show that DACC is strategy-proof for a subset of agents if and only if it generates stable matchings that are most preferred (among all stable matchings) by each agent in that subset. Therefore, the question of strategy-proofness for a subset of agents boils down to understanding the mapping between orders Φ and the corresponding stable matchings. As noted in the preceding paragraph, this mapping is probably complicated.

On Fairness

Fairness is an important concern in practical market design. Because the DACC algorithm doesn't differentiate between the two sides of the market, it has many "fair" implementations. The sequence of proposers can be chosen i.i.d. uniformly at random. Alternatively, agents can be randomly ordered, and the fixed order repeated ad infinitum. The flexibility of DACC allows for further fine-tuning; for example, the order can be reversed so that the agent who applied last, applies first in the next block of proposals. Compared to a fair randomization over the men- and women-proposing 1DA, DACC lowers the variance of outcomes (measured by the rank of the stable-match partner), as it often produces non-extreme stable matchings.

Teo and Sethuraman (1998) and Schwarz and Yenmez (2011) define a median matching, which can be viewed as ex-post "fair", and which reduces the variance even further. Cheng (2008) shows that finding a median matching is NP-hard in certain instances, while running DACC for a given sequence of proposers requires polynomial time. Moreover, as argued by Li (2015), in practical market design, desirable properties of mechanisms often matter to the extent that they are recognized by participating agents. DACC is "procedurally fair", i.e. understanding that it is "fair" does not require understanding the exact mapping from inputs to outcomes.

The algorithms proposed by Ma (1996) and Romero-Medina (2005) also achieve procedural, or ex-ante, fairness, and are easy to implement and understand. Ex-ante fair matching algorithms are analyzed by Klaus and Klijn (2006) and Kuvalekar (2015), although these are more structural in nature. To the best of my knowledge, none of the above algorithms generates all stable matchings in the one-to-one setting. Moreover, DACC admits a natural extension to many-to-one matching with contracts (see Appendix D) which is necessary for most practical applications.

6 Can DACC be made simpler?

The class of DACC algorithms has three main properties, proved in Section 3: *(1)* every DACC algorithm stops in finite time, *(2)* if a DACC algorithm stops, the outcome is stable, and *(3)* every stable matching can be achieved by an algorithm from the DACC class. In this section, I explore the potential of simpler classes of mechanisms to achieve properties 1-3. I define two natural simplifications of DACC, the Two-Sided Deferred Acceptance (2DA) algorithms and the Budget-Based Two-Sided Deferred Acceptance (B2DA) algorithms, and demonstrate that:

- 1DA has property 1 and 2, but not 3;
- 2DA has property 1 and 3, but not 2;

- B2DA has property 2 and 3, but not 1.

Thus, none of the features of DACC (in particular the presence of CCs) are redundant.

6.1 1DA (Gale-Shapley algorithm)

Gale and Shapley (1962) proved that 1DA algorithms have property 1 and 2. Property 3 fails because there generally exist stable matchings that are neither men- nor women-optimal. In Appendix B, I present a simple example of a market with three stable matchings and demonstrate how the median matching can be achieved by DACC.

6.2 2DA (Two-Sided Deferred Acceptance)

In the Two-Sided Deferred Acceptance algorithm, the order in which agents make offers is still governed by Φ. Whenever it's i's turn to apply, i applies to the best partner that hasn't rejected or divorced i yet (effectively, budget sets are replaced by rejection sets that can only grow). We do not run CCs. The algorithm terminates when every agent is matched to the best partner among those who haven't rejected or divorced him or her (or unmatched if rejected by all acceptable partners).

2DA stops in finite time due to monotonicity of rejection sets. The proof of Claim 4 shows that every stable matchings can be achieved by 2DA with an appropriately chosen Φ. However, there are matching markets and sequences Φ for which 2DA will converge to an unstable outcome. An example is provided in Appendix B. The gist of the example is as follows. Suppose that i and j should be matched at a stable outcome. When i proposes to j, j is temporarily matched to a more preferred partner, and hence rejects i. Later, j loses this better match, and proposes to i who is now matched to a more preferred partner, and hence rejects j. This double deviation can occur because, unlike in 1DA, offers can be withdrawn when both sides of the market propose.

6.3 B2DA (Budget-Based Two-Sided Deferred Acceptance)

The Budget-Based Two-Sided Deferred Acceptance algorithm corrects the double-rejection problem of 2DA by replacing rejection sets with budget sets. Formally, B2DA is defined as DACC, but without the compensation chains.

Because j is added to i's budget set when j applies to i, a double rejection does not prevent i and j from being matched to each other when the algorithm stops. The proof of Claim 1 applies directly to B2DA giving property 2, and the proof of Claim 4 establishes property 3. The price we pay is lack of property 1. Unlike rejection sets, budget sets behave in a non-monotone way. In the absence of CCs, it is possible to construct a matching market and a Φ such that the B2DA algorithm falls into a loop. Budget sets fluctuate, and proposals and acceptance decisions exhibit a recurring pattern. In the language of Section 4, B2DA does not admit a potential function. Details are provided in Appendix B.

7 Concluding Remarks

This paper establishes an equivalence between stability and a class of deferred acceptance procedures by defining a generalization of the Gale-Shapley algorithm in which both sides of the market may propose in an arbitrary pre-determined order. For every order in which agents are allowed to make offers, the DACC algorithm converges in finite time to a stable matching, and every stable matching can be obtained by appropriately choosing the sequence of proposers.

The DACC algorithms have attractive properties from a practical market design perspective. They always reach a stable outcome and are relatively easy to understand and implement. They can treat the two sides of the market symmetrically, for example, if the sequence of proposers is chosen uniformly at random. Unlike some other algorithms with above features, they generalize easily to many-to-one matching with contracts which is important from the point of view of practical applicability.

Because the definition of DACC does not rely on the two-sidedness of the marriage market, it can be applied to the roommates problem (see Appendix C). It would be interesting to see if DACC could work equally well in other settings, e.g. the coalition formation problem (see Pycia, 2012).

The DACC algorithms shed some light on the behavior of decentralized matching markets where offers may be made in an arbitrary (random) order. Just as the tâtonnement process for prices provides theoretical support for convergence of markets to the competitive equilibrium, DACC establishes sufficient conditions (complementary to Roth and Vande Vate, 1990) for decentralized matching markets to reach stability.

References

ABDULKADIROĞLU, A., PATHAK, P. A. and ROTH, A. E. (2005a). The New York City High School Match. *American Economic Review*, **95** (2), 364–367.

—, —, ROTH, A. E. and SÖNMEZ, T. (2005b). The Boston Public School Match. *American Economic Review*, **95** (2), 368–371.

ADACHI, H. (2000). On a characterization of stable matchings. *Economics Letters*, **68**, 43–49.

ARROW, K. J. and HURWICZ, L. (1958). On the Stability of the Competitive Equilibrium, I. *Econometrica*, **26** (4), 522–552.

AUSUBEL, L. M. (2006). An Efficient Dynamic Auction for Heterogeneous Commodities. *American Economic Review*, **96** (3), 602–629.

AYGÜN, O. and SÖNMEZ, T. (2013). Matching with contracts: Comment. *American Economic Review*, **103** (5), 2050–51.

AZEVEDO, E. M. and HATFIELD, J. W. (2015). *Existence of Equilibrium in Large Matching Markets with Complementarities*. Working paper.

BLUM, Y., ROTH, A. E. and ROTHBLUM, U. G. (1997). Vacancy chains and equilibrium in senior-level labor markets. *Journal of Economic Theory*, **76**, 362–411.

— and ROTHBLUM, U. G. (2002). 'Timing Is Everything' and Marital Bliss. *Journal of Economic Theory*, **103** (2), 429–443.

CHE, Y.-K., KIM, J. and KOJIMA, F. (2015). *Stable Matching in Large Economies*. Working paper.

CHENG, C. T. (2008). The Generalized Median Stable Matchings: Finding Them Is Not That Easy. In E. S. Laber, C. Bornstein, L. T. Nogueira and L. Faria (eds.), *LATIN 2008: Theoretical Informatics*, no. 4957 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 568–579.

CHUNG, K.-S. (2000). On the Existence of Stable Roommate Matchings. *Games and Economic Behavior*, **33** (2), 206–230.

ECHENIQUE, F. and OVIEDO, J. (2006). A theory of stability in many-to-many matching markets. *Theoretical Economics*, **1** (2), 233–273.

FLEINER, T. (2003). A Fixed-point Approach to Stable Matchings and Some Applications. *Math. Oper. Res.*, **28** (1), 103–126.

GALE, D. and SHAPLEY, L. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, **69** (1).

Gudmundsson, J. (2013). *A Review of the Existence of Stable Roommate Matchings*. Working Papers 8, Lund University, Department of Economics.

Gul, F. and Stacchetti, E. (1999). Walrasian Equilibrium with Gross Substitutes. *Journal of Economic Theory*, **87** (1), 95–124.

Hatfield, J. W. and Milgrom, P. R. (2005). Matching with contracts. *American Economic Review*, **95** (4), 913–935.

Irving, R. W. (1985). An efficient algorithm for the stable roommates problem. *Journal of Algorithms*, **6** (4), 577–595.

— and Leather, P. (1986). The Complexity of Counting Stable Marriages. *SIAM J. Comput.*, **15** (3), 655–667.

Kelso, A. S. and Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes. *Econometrica*, **50** (6), 1483–1504.

Kesten, O. (2004). A New Solution to Probabilistic Stability: The Compromise Mechanism, unpublished Manuscript.

Klaus, B. and Klijn, F. (2006). Procedurally fair and stable matching. *Economic Theory*, **27** (2), 431–447.

Knuth, D. E. (1976). Mariages stables. *Les Presses de l'Universite de Montreal*.

Kojima, F. and Ünver, M. U. (2008). Random paths to pairwise stability in many-to-many matching problems: a study on market equilibration. *International Journal of Game Theory*, **36** (3), 473–488.

Kuvalekar, A. V. (2015). *A Fair Algorithm in Marriage Market*. Working paper.

Li, S. (2015). *Obviously Strategy-Proof Mechanisms*. Working paper.

Ma, J. (1996). On Randomized Matching Mechanisms. *Economic Theory*, **8** (2), 377–81.

Martínez, R., Massó, J., Neme, A. and Oviedo, J. (2004). An algorithm to compute the full set of many-to-many stable matchings. *Mathematical Social Sciences*, **47** (2), 187 – 210.

McVitie, D. G. and Wilson, L. B. (1971). Three Procedures for the Stable Marriage Problem. *Commun. ACM*, **14** (7), 491–492.

Pycia, M. (2012). Stability and Preference Alignment in Matching and Coalition Formation. *Econometrica*, **80** (1), 323–362.

Romero-Medina, A. (2005). Equitable Selection in Bilateral Matching Markets. *Theory and Decision*, **58** (3), 305–324.

Roth, A. E. (2007). *Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions*. Nber working paper 13225.

— and Vande Vate, J. H. (1990). Random paths to stability in two-sided matching. *Econometrica*, **58** (6), pp. 1475–1480.

Schwarz, M. and Yenmez, M. B. (2011). Median stable matching for markets with wages. *Journal of Economic Theory*, **146** (2), 619–637.

Sönmez, T. (1999). Strategy-Proofness and Essentially Single-Valued Cores. *Econometrica*, **67** (3), 677–689.

Sönmez, T. and Switzer, T. B. (2013). Matching With (Branch-of-Choice) Contracts at the United States Military Academy. *Econometrica*, **81** (2), 451–488.

Sun, N. and Yang, Z. (2009). A Double-Track Adjustment Process for Discrete Markets With Substitutes and Complements. *Econometrica*, **77** (3), 933–952.

Teo, C.-P. and Sethuraman, J. (1998). The Geometry of Fractional Stable Matchings and Its Applications. *Math. Oper. Res.*, **23** (4), 874–891.

Uzawa, H. (1960). Walras' Tâtonnement in The Theory of Exchange. *The Review of Economic Studies*, **27** (3), 182–194.

Wirth, N. (1978). *Algorithms + Data Structures = Programs*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

A Appendix A - Proof of Lemma 1

I let k index the rounds in the DACC algorithm, and I use the superscript k to denote sets at the end of round k. For example, A_i^k is the set of agents that applied to agent i up to and including round k.

First, note that the sets A_i^k never shrink. Thus, for a fixed Φ, there exists a round $k^\star$ such that all A_i^k are constant after $k^\star$. For all $k \geq k^\star$, define the set X^k as

$$X^k = \{\{i,\, j\} : i \in N,\, j \in N_i,\;\; i \text{ and } j \text{ never interact after round } k\}.$$

Moreover, to simplify notation, let $d_i^k = d_i(\mathcal{B}^k,\, \mu^k)$, and $d^k = \sum_{i \in N} d_i^k$.

Claim 5. *For every $k > k^\star$, unless $d_{\Phi(k)}^{k-1} = 0$, either d^k decreases strictly or $|X^k|$ grows strictly.*

Proof. Fix a round $k > k^\star$. If $d^{k-1}_{\Phi(k)} = 0$, then $\Phi(k)$ is already matched to the most preferred partner in his or her budget set, and thus nothing happens in round k. I assume from now on that $d^{k-1}_{\Phi(k)} > 0$ which means that $\Phi(k)$ proposes in round k.

I prove that the only case in which d^k doesn't go strictly down relative to d^{k-1} is when some agent l is divorced by some $l' \notin A^k_l$ in round k. That is, if either (i) there are no divorces in round k, or (ii) all divorces lead to CCs, then d^k decreases strictly in that round.

Denote by j the agent that $i = \Phi(k)$ proposes to. By direct inspection, $d^k_i + d^k_j$ goes down strictly regardless of whether i's offer is rejected or accepted. If i and j were not matched to anyone, there are no divorces. This is case (i). Otherwise, we have to show that the function d decreases weakly along a CC. That is, the value it takes when some agent l is divorced (and we run a CC starting at l) is not smaller that the value it takes when this CC stops. This will cover case (ii).

Suppose wlog that l is a man. Then, in the CC starting at l, women receive offers, so $\sum_{w\in W} d_w$ decreases weakly along the CC. By definition of a CC, all men who apply in a CC end up being matched to the most preferred option in their respective budget sets. Thus, $d^k_m = 0$ for all m who apply in the CC, and $\sum_{m\in M} d_m$ must decrease at least weakly as well.

Now suppose that d^k doesn't strictly decrease in round k. By what I have shown so far, it must be that some agent l is divorced by $l' \notin A^k_l$, i.e. we have a divorce which is not followed by a CC. Because l is divorced, we have $l' \notin B^k_l$. Because $l' \notin A^k_l$ and $A^k_l = A^{k+n}_l$ for any $n \in \mathbb{N}$ (because $k > k^\star$), $l' \notin B^{k+n}_l$ for all $n \in \mathbb{N}$. That is, l can never apply to l'. And due to $l' \notin A^{k+n}_l$ for all n, l' never applies to l either. Thus, we add $\{l, l'\}$ to X^k, and thus $|X^k|$ grows strictly. $\square$

To finish the proof, I show how to choose the sequence a. Because $|X^k|$ is bounded above by the number of potential pairs of agents, $|X^k| - |X^{k-1}| > 0$ in only finitely many rounds k. Thus, there exists $\bar{k} > k^\star$ such that $|X^k|$ is constant after $\bar{k}$.

By Claim 5, in all rounds $k > \bar{k}$, either $d^{k-1}_{\Phi(k)} = 0$ (in which case nothing happens and d^k stays constant), or d^k decreases strictly. I define a recursively starting from $a_0 = \bar{k}$. Having picked $(a_0, a_1, ..., a_{n-1})$, and assuming that the algorithm hasn't stopped at a_{n-1}, define

$$a_n = \min\{k \in \mathbb{N} :\ k > a_{n-1},\, d^{k-1}_{\Phi(k)} > 0\}.$$

The number a_n is well defined. Indeed, because the algorithm didn't stop at a_{n-1}, there exists an agent i with $d^{a_{n-1}}_i > 0$. By assumption, Φ takes the value i infinitely many times, and thus $\Phi(k) = i$ for some $k > a_{n-1}$. Having excluded rounds in which d^k stays constant, we know that d decreases strictly along the sequence $(\mathcal{B}^{a_n},\, \mu^{a_n})_{n=1,2,...}$.

B Appendix B - Examples

This section presents three examples corresponding to the three subsections of Section 6. In all examples, the market consists of three agents on each side, i.e. $M = \{m_1, m_2, m_3\}$ and $W = \{w_1, w_2, w_3\}$. When introducing preferences, I list only acceptable partners.

B.1 Example for Subsection 6.1

The example below shows the simplest possible matching market with a stable matching that is neither men- nor women-optimal. This example is known, but I present it here for completeness, and to illustrate how the DACC algorithm works.

Example 1. 1DA does not generally achieve all stable matchings.

Let the preferences be given by:

$$
\begin{array}{ll}
m_1 : w_1 \succ w_2 \succ w_3 & \qquad w_1 : m_2 \succ m_3 \succ m_1 \\
m_2 : w_2 \succ w_3 \succ w_1 & \qquad w_2 : m_3 \succ m_1 \succ m_2 \\
m_3 : w_3 \succ w_1 \succ w_2 & \qquad w_3 : m_1 \succ m_2 \succ m_3
\end{array}
$$

There are three stable matchings: men-optimal $\mu^M = \{\{m_1, w_1\}, \{m_2, w_2\}, \{m_3, w_3\}\}$, women-optimal $\mu^W = \{\{m_1, w_3\}, \{m_2, w_1\}, \{m_3, w_2\}\}$, and the median matching $\mu^\star = \{\{m_1, w_2\}, \{m_2, w_3\}, \{m_3, w_1\}\}$. The median matching cannot be achieved by 1DA.

To see how the DACC algorithm can lead to $\mu^\star$, consider the sequence

$$\Phi = m_1, w_1, m_2, w_2, m_3, w_3, m_1, m_2, m_3, \ldots.$$

In the first six rounds, all agents get a chance to propose to their favorite partner, and subsequently get divorced. Thus, starting from round 7, all agents have budget sets without their most preferred partners. In rounds 7-9, men propose to their second choices, and we reach $\mu^\star$.

B.2 Example for Subsection 6.2

In this subsection, I demonstrate how 2DA may fail to achieve a stable matching.

Example 2. 2DA may in general converge to an unstable outcome.

Consider the following preferences:

$$m_1 : w_3 \succ w_1 \qquad w_1 : m_2 \succ m_1$$
$$m_2 : w_2 \succ w_1 \qquad w_2 : m_3 \succ m_2$$
$$m_3 : w_3 \succ w_2 \qquad w_3 : m_3 \succ m_1$$

The unique stable matching is $\mu^\star = \{\{m_1, w_1\}, \{m_2, w_2\}, \{m_3, w_3\}\}$. Consider a sequence Φ with initial ordering over agents as shown in Table 1. Then, in the matching achieved by 2DA, m_1 and w_1 are unmatched, contradicting stability (see Table 1 for details).

Tab. 1: 2DA - failure of stability

Round k	$\Phi(k)$	applies to	Decision	Current matches
1	w_1	m_2	accept	w_1m_2
2	m_2	w_2	accept	$\cancel{w_1m_2}$, m_2w_2
3	m_1	w_3	accept	m_2w_2, m_1w_3
4	w_1	m_1	reject	m_2w_2, m_1w_3
5	w_2	m_3	accept	$\cancel{m_2w_2}$, m_1w_3, w_2m_3
6	m_2	w_1	accept	m_1w_3, w_2m_3, m_2w_1
7	w_3	m_3	accept	$\cancel{m_1w_3}$, $\cancel{w_2m_3}$, m_2w_1, w_3m_3
8	m_1	w_1	reject	m_2w_1, w_3m_3
9	w_2	m_2	accept	$\cancel{m_2w_1}$, w_3m_3, w_2m_2
10	m_1	$\emptyset$	accept	w_3m_3, w_2m_2, $m_1\emptyset$
11	w_1	$\emptyset$	accept	w_3m_3, w_2m_2, $m_1\emptyset$, $w_1\emptyset$
...	...	...	...	...

The algorithm fails to produce a stable matching because when w_1 applies to m_1 in round 4, m_1 is temporarily matched to a more preferred w_3. By the time when m_1 applies to w_1 in round 8, w_1 is temporarily matched to a preferred m_2. As a consequence, m_1 and w_1 reject each other although they should be matched in the unique stable matching.

Suppose DACC were run instead of 2DA for the same sequence Φ. Then, in round 9, w_1 is compensated because w_1 is divorced by m_2 who applied to w_1 in round 6. Because $m_1 \in B_{w_1}$ (m_1 proposed to w_1 in round 8), and $m_2 \notin B_{w_1}$ (m_2 divorced w_1), w_1 proposes to m_1 and the stable matching is reached.

B.3 Example for Subsection 6.3

Finally, I show that in the absence of CCs, the two-sided deferred acceptance procedure may never stop.

Example 3. B2DA may never converge.

Consider the following preferences:

$$m_1 : w_3 \succ w_2 \qquad w_1 : m_3 \succ m_2$$
$$m_2 : w_1 \succ w_3 \qquad w_2 : m_1 \succ m_3$$
$$m_3 : w_2 \succ w_1 \qquad w_3 : m_2 \succ m_1$$

Take the sequence $\Phi = w_2, m_2, m_3, w_3, (m_3, w_3, m_2, w_2, m_1, w_1), ...,$ where the string in brackets is repeated periodically. Table 2 illustrates how the B2DA algorithm falls into a loop.

Tab. 2: B2DA - failure of convergence

Round k	$\Phi(k)$	applies to	Changes in budgets	Current matches
1	w_2	m_1	-	w_2m_1
2	m_2	w_1	-	w_2m_1, m_2w_1
3	m_3	w_2	$w_2 \notin B_{m_3}$	w_2m_1, m_2w_1
4	w_3	m_2	$m_2 \notin B_{w_3}$	w_2m_1, m_2w_1
5 (+6n)	m_3	w_1	$m_3 \in B_{w_1}$, $w_1 \notin B_{m_2}$	w_2m_1, $\cancel{m_2w_1}$, m_3w_1
6 (+6n)	w_3	m_1	$w_3 \in B_{m_1}$, $m_1 \notin B_{w_2}$	$\cancel{w_2m_1}$, m_3w_1, w_3m_1
7 (+6n)	m_2	w_3	$m_2 \in B_{w_3}$, $w_3 \notin B_{m_1}$	m_3w_1, $\cancel{w_3m_1}$, m_2w_3
8 (+6n)	w_2	m_3	$w_2 \in B_{m_3}$, $m_3 \notin B_{w_1}$	$\cancel{m_3w_1}$, m_2w_3, w_2m_3
9 (+6n)	m_1	w_2	$m_1 \in B_{w_2}$, $w_2 \notin B_{m_3}$	m_2w_3, $\cancel{w_2m_3}$, m_1w_2
10 (+6n)	w_1	m_2	$w_1 \in B_{m_2}$, $m_2 \notin B_{w_3}$	$\cancel{m_2w_3}$, m_1w_2, w_1m_2

The reason why convergence may fail is easy to understand when we compare B2DA with 1DA. In the men-proposing DA, budget sets of women can only increase, and budget sets of men can only decrease. Due to this monotonicity, the 1DA algorithm always converges. In the B2DA, budget sets of agents may change in both directions. This is the case in the example. During the cycle, each agent i receives an application from the most preferred partner j, and thus rejects the current partner. But then j receives as application from j's most preferred partner, and thus divorces i, and so on. The budget sets fluctuate accordingly.

Suppose we ran DACC with the same sequence of applicants. The initial 6 steps are identical. In round 7, w_3 divorces m_1 because she receives a better offer from m_2. At that time, $w_3 \in A_{m_1}$ because w_3 applied to m_1 in round 6. Thus, we start a CC at m_1. We have $B_{m_1} = \{w_1, w_2\}$, so m_1 applies to w_2. Woman w_2 accepts the offer and a stable matching is reached.

Online Appendix (Not For Publication)

C Appendix C - The Stable Roommates Problem

In this appendix, I discuss the application of the DACC algorithm to the stable roommates problem (henceforth SRP). That is, agents are allowed to have preferences over the entire set of agents N, and a matching is a partition of N into sets of cardinality at most two.[16] I continue to assume for ease of exposition that preferences are strict. The definition of DACC made no reference to the two-sidedness of the marriage market and thus it applies directly to the SRP.

As observed by Gale and Shapley (1962), a stable matching may fail to exist for the SRP. Because Claim 1 remains true, DACC cannot be guaranteed to stop. This is due to failure of Claim 2 - in the absence of the two-sided structure the chain of compensations may never end. However, if all CCs stop in finite time, the DACC algorithm converges.

Observation 4. *For an instance of the SRP and a given sequence Φ, if all CCs stop in finite time, then DACC converges in finite time to a stable matching.*

The observation follows directly from the proofs in Section 3.

If stable matchings exist, they can all be achieved by DACC algorithms.

Proposition 2. *For any stable matching in the SRP, there is an ordering Φ such that μ is the outcome of DACC(Φ). Moreover, it can be achieved with an order Φ that does not lead to any CCs.*

The proof is identical to that of Claim 4, except that a separate argument is needed to show convergence of DACC for the constructed order over applying agents. Because there are no CCs under that order, this follows directly from Observation 4.

If all CCs stop in finite time under a sequence Φ, then by Observation 4, a stable matching exists in the SRP. More primitive sufficient conditions for existence of stable matchings in the SRP can be found in the literature.

Definition 1. Let L be an ordered list of $k \geq 3$ agents in the SRP. L is a ring with respect to strict preferences $\succ$ if

$$\forall i \in L,\ L(i+1) \succ^{L(i)} L(i-1),$$

where all indices are taken modulo k.

[16] See Gudmundsson (2013) for a formal definition of the SRP.

Chung (2000) shows that if there are no odd rings, then a stable matching exists for the SRP.

Proposition 3. *If there is no odd ring in the SRP, then for any sequence Φ, the DACC(Φ) algorithm converges in finite time to a stable matching.*

The proposition follows directly from the results in Section 3 and the lemma below whose proof I provide in Subsection C.1. Existence of a stable matching is obtained as a trivial corollary.

Lemma 2. *Fixing Φ and an instance of the SRP, if a compensation chain never stops, then there exists an odd ring among (a subset of) the agents participating in that CC.*

It is easy to construct examples in which a stable matching exists despite the presence of odd rings, and where a CC never stops in the DACC algorithm. Irving (1985) provides an algorithm that finds a stable matching whenever it exists, regardless of the properties of preferences.

C.1 Proof of Lemma 2

Suppose that a CC never stops. Because there is a finite number of agents, the CC must cycle, i.e. there is a sequence of proposals (and corresponding acceptance/ rejection decisions) that repeats itself indefinitely. Note that once the CC enters a cycle, each agent, whenever matched, is matched to the best option in his or her budget set.

Fixing a full cycle of the CC, for each agent i, the following event must take place within the cycle: i applies, i's application is accepted, and later i receives an application that i accepts (before applying again). Let n_i be the number of rounds between the successful application of i, and the moment when i receives the offer, as described above. If there are several such events for i within the cycle, take n_i to be the minimum.

I prove that such an event indeed takes place for each i. There exists a round in the cycle of the CC when i has the smallest budget set (among budget sets that i has during the cycle). Since the smallest budget set must arise after i is rejected or divorced, there exists a round when i applies under this smallest budget set. The application of i must be accepted, as otherwise the budget set would shrink further. Moreover, and for the same reason, i must subsequently receive an application that is accepted before i applies again.

Let j be the agent with the smallest n_i across all i. Let $a_0 = j$, and consider the round in the cycle when j applies under the smallest budget set. Let a_1 be the agent who accepts the application of j. Because the CC continues, a_1 must have been matched to some a_2. Moreover, a_1 prefers j to a_2, and a_2 applies next. Note that the length of the sequence $a = (a_0,\, a_1,\, a_2)$ is an odd number, and a_1 prefers a_0 to a_2.

Agent a_2 is eventually accepted (otherwise the CC would stop) by some agent a_3. By the observation made in the first paragraph, a_2 prefers a_1 to a_3. a_3 was matched to some a_4, and a_3 prefers a_2 to a_4. The sequence a is of odd length, and for all $l = 1,\, 2,\, 3$, a_l prefers a_{l-1} to a_{l+1}.

Proceeding inductively, we get a sequence $a = (a_0,\, a_1, \ldots a_{2m+1})$ such that for all $l = 1,\, 2, \ldots,\, 2m$, a_l prefers a_{l-1} to a_{l+1}. There exists a smallest $\bar{m}$ such that $a_{2\bar{m}+1}$ applies to $a_0 = j$. a_0 prefers $a_{2\bar{m}+1}$ to a_1 by the observation made in the first paragraph.

For $a = (a_0,\, a_1, \ldots a_{2\bar{m}+1})$ to be an odd ring, it would have to be that no agent appears twice in a. There cannot be an agent who applies and then receives an application as that would contradict the choice of j as the agent with the smallest n_j. So suppose that there is an agent k who applies (and gets accepted) twice in a. That is, we have a subsequence $b = (a_{2l},\, a_{2l+1}, \ldots,\, a_{2m})$ of a such that $a_{2l} = a_{2m} = k$. Then we can delete the subsequence $(a_{2l+1}, \ldots,\, a_{2m})$ from a without changing its properties. In particular, a remains to be of odd length. Deleting such subsequences iteratively, we eventually arrive at an odd ring.

D Appendix D - DACC for Matching with Contracts

D.1 The model

I adopt the framework for matching with contracts from Hatfield and Milgrom (2005). The results of Aygün and Sönmez (2013) allow me to use choice functions (as opposed to preference relations) as a primitive description of agents' preferences.[17]

Let D be the set of doctors, H the set of hospitals, and X the set of contracts. Each $x \in X$ is a bilateral contract between a doctor $d_x \in D$ and a hospital $h_x \in H$. When agent i signs contract x, i^c_x denotes the counterparty of i under contract x. X always contains the null contract $\emptyset$ that can be chosen unilaterally by any agent.

Each doctor $d \in D$ can sign at most one contract, and the contract must be from the set $X_d \equiv \{x \in X : d_x = d\}$. Preferences of doctor d are given by a choice function $C_d : 2^X \to 2^X$ which for every subset of contracts $Y \subset X$ returns a single contract that d prefers most among $Y \cap X_d$, or $\emptyset$ if there are no acceptable contracts in $Y \cap X_d$.

Each hospital $h \in H$ can sign multiple contracts from $X_h \equiv \{x \in X : h_x = h\}$

[17] Aygün and Sönmez (2013) argue that the results of Hatfield and Milgrom (2005) depend only on the properties of choice functions derived from the primitive preference relations. They show that existence of an underlying preference relation is an unnecessarily strong assumption. They propose to define choice functions as a primitive of the model under the assumption that choice functions satisfy *irrelevance of removed contracts.*

but at most one with any given doctor $d \in D$. The preferences of hospital h are given by a choice function $C_h : 2^X \to 2^X$ which for every $Y \subset X$ returns the set of chosen contracts, a subset of $Y \cap X_h$. I assume throughout that C_h, for every $h \in H$, satisfies *irrelevance of rejected contracts*.

Definition 2 (Aygün and Sönmez, 2013). Contracts satisfy irrelevance of rejected contracts (IRC) for hospital h if

$$\forall Y \subset X, \forall z \in X \setminus Y, z \notin C_h(Y \cup \{z\}) \implies C_h(Y) = C_h(Y \cup \{z\}).$$

By induction, one can show that IRC is equivalent to the following seemingly stronger property that is more convenient to work with.

Definition 3. Contracts satisfy strong irrelevance of rejected contracts (SIRC) for hospital h if

$$\forall Y \subset X, \forall Z \subset X \setminus Y, Z \cap C_h(Y \cup Z) = \emptyset \implies C_h(Y) = C_h(Y \cup Z).$$

For a set of contracts Y, let Y_i denote the subset of contracts in Y that name agent i as a signee. I say that a set of contracts $Y_i \subset X_i$ is acceptable for i if $C_i(Y_i) = Y_i$.

Definition 4 (Hatfield and Milgrom, 2005). A set of contracts $Y^\star \subset X$ is a stable allocation (stable set of contracts) if

1. $Y_i^\star$ is acceptable for i, for each $i \in H \cup D$,
2. There exists no hospital h and set of contracts $Y \neq C_h(Y^\star)$ such that

$$Y = C_h(Y^\star \cup Y) \subset \bigcup_{d \in D} C_d(Y^\star \cup Y).$$

Finally, a substituability condition on hospitals' preferences is needed to obtain existence of stable allocations, as shown by Kelso and Crawford (1982) or Hatfield and Milgrom (2005). For any $i \in H \cup D$, let $R_i(Y) = Y \setminus C_i(Y)$ be the set of contracts rejected from the set Y by agent i.

Definition 5. Contracts in X are substitutes for hospital h, if for all subsets $Y' \subset Y \subset X$, $R_h(Y') \subset R_h(Y)$.

D.2 Definition of DACC

There are multiple ways to generalize DACC to matching with contracts. I propose a definition that aims to reflect the decentralized and sequential nature of DACC by assuming that the proposer, whether a doctor or a hospital, can propose at most one contract at a time. The definition is equivalent to the baseline definition of DACC from Section 3 if hospitals have singleton preferences.

Every agent i starts with a full budget set $B_i = X_i$ of all contracts that i can potentially sign. The initial set of signed contracts Y is empty. The budget system $\{B_i\}_{i \in H \cup D}$ and the set Y are adjusted during the course of the algorithm.

Proposals and Acceptance. In round k, agent $i = \Phi(k)$ proposes a contract $x \in C_i(B_i) \setminus Y_i$ to $j = i^c_x$.[18] The choice of x in case $C_i(B_i) \setminus Y_i$ has more than one element is arbitrary and will not influence subsequent results. Agent j (tentatively) signs the contract if $x \in C_j(Y_j \cup \{x\})$. In that case, we adjust Y by adding x, and removing contracts $R_j(Y_j \cup \{x\})$ and $R_i(Y_i \cup \{x\})$. Otherwise, j rejects x and the allocation Y is unchanged.

Budget Sets. Whenever some agent i proposes a contract x to $j = i^c_x$, we add x to j's budget set B_j. Whenever x is rejected or broken by i, we remove x from the budget set B_j of $j = i^c_x$. Throughout, the term "contract x is broken by i" means that i and i^c_x signed contract x, and then i broke the contract with i^c_x after signing another contract.

Compensation Chains (CCs). In the current setting, CCs are more complicated because multiple contracts can be broken simultaneously, leading to multiple deceptions. (A more accurate name for a CC would be a compensation *cascade*.) To address this issue, I introduce a notion of a waitlist. The waitlist contains all agents waiting to be compensated in the current CC. A formal definition of a CC is simplified by using the potential function to describe it. I define the potential function as

$$\boldsymbol{d}_i(B_i, Y_i) = |\{x \in B_i \setminus Y_i : x \in C_i(Y_i \cup \{x\})\}|. \tag{D.1}$$

That is, $\boldsymbol{d}_i$ counts the number of additional contracts agent i would accept given her current allocation Y_i. This definition coincides with (3.1) when agent i can sign at most one contract.

I say that i *deceived* $j = i^c_x$ if i broke the contract x which i proposed before. When some i deceives j, we compensate agent j (with one exception, described below).

Compensation takes the following form. Let d^0_j be the value of j's potential function (D.1) prior to deception. Agent j is allowed to propose contracts, regardless of the sequence Φ, until her $\boldsymbol{d}_j$ falls weakly below d^0_j. This may in general require multiple proposals (unlike in the one-to-one case, where one accepted proposal brings $\boldsymbol{d}_j$ to zero).

During j's compensation, whenever a proposed contract is accepted by k who by doing so deceives her counterparty k^c_y for some contract y, we put k^c_y on the waitlist.

[18] If $C_i(B_i) \subseteq Y_i$, we skip the round.

After j is compensated, we compensate the next agent on the waitlist (the order does not matter). Agents are removed from the waitlist after being compensated.

The CC ends when the last agent on the waitlist is compensated, and no new agents are added to the waitlist. Then, the algorithm proceeds to the next round, and the proposer is determined by Φ.

One type of deception does not induce compensation. Within a CC, if an agent i deceives agent j as a consequence of i proposing a contract (as opposed to i receiving an offer), the deceived agent j is **not** compensated. Such a scenario is possible when an agent holds multiple contracts, proposes a new one which is accepted, and as a consequence breaks some of the old contracts. The counterparties of the old contracts are not compensated when this takes place inside a CC. This has no bite in the one-to-one case where the agent proposing in a CC is always unmatched. In the many-to-one setting, this restriction ensures that in a CC initiated at a hospital, only hospitals are compensated (the analogous property holds automatically for doctors).

The algorithm stops when $Y_i = C_i(B_i)$ for all agents i.

D.3 Convergence to a Stable Allocation

The following result generalizes the first part of Theorem 1.

Theorem 2. *If contracts are substitutes for hospitals, then for any sequence Φ, $DACC(\Phi)$ stops in finite time and its outcome Y is a stable allocation.*

The proof of Theorem 2 is similar to the proof of Theorem 1, and I skip the parts that are analogous. Several additional arguments are needed to fill in the details, and I provide them below.

First, if DACC stops, the outcome is a stable allocation (see Claim 1). The proof of Claim 1 can be translated directly to the language of matching with contracts, and there are no substantial changes to the argument. The current set of contracts remains acceptable for agents when contracts are broken because of the substitutes property.

Second, every CC stops in finite time (see Claim 2). This follows from the observation made above that in a CC initiated at a hospital (doctor), only hospitals (doctors) apply. Budget sets can never grow inside a CC for the proposing side. Compensation for any agent i ends, because as i applies, her potential function $\boldsymbol{d}_i$ goes strictly down.[19] After every compensation, either some agent's budget set on the proposing side shrinks, or the waitlist's length is reduced. Because budget sets cannot decrease indefinitely, the waitlist will eventually be empty.

[19] In the case when a proposed contract is accepted by i's counterparty, this conclusion requires that contracts are substitutes.

Third, DACC stops in finite time (see Claim 3). To prove that, I follow the approach from the one-to-one case by showing that $\boldsymbol{d}_i$ is a potential function.

Lemma 3. *When contracts are substitutes, $\boldsymbol{d}_i(B_i, Y_i) = 0$ if and only if $C_i(B_i) = Y_i$.*

Proof. Suppose that $\boldsymbol{d}_i(B_i, Y_i) = 0$. Let $A := C_i(B_i)$. Because $Y_i \subset B_i$, by SIRC, $A = C_i(Y_i \cup A)$. Towards a contradiction, suppose that $A \neq Y_i$. Because Y_i is acceptable, it cannot be that $A \subsetneq Y_i$, so there exists $x \in A \setminus Y_i$. Due to $\boldsymbol{d}_i(B_i, Y_i) = 0$, we have $x \in R_i(Y_i \cup \{x\})$. Then, by substitutes, $x \in R_i(Y_i \cup A)$, a contradiction with $A = C_i(Y_i \cup A)$.

In the opposite direction, suppose that $C_i(B_i) = Y_i$. If $\boldsymbol{d}_i(B_i, Y_i) > 0$, then there exists $x \in B_i \setminus Y_i$, $x \in C_i(Y_i \cup \{x\})$. That contradicts SIRC. □

Next, it is easy to show that $\boldsymbol{d}_i$ goes strictly down when i's offer is rejected, or when i signs a new contract (the latter property requires substitutes). By the same argument as in the proof of Claim 3, for every hospital $\boldsymbol{d}_i$ will eventually converge to zero. An important part of that argument was that after sufficiently many rounds, every time a contract is broken, it is a deception, and hence a CC follows.

However, this argument is not enough to establish an analogous property for doctors. Hospitals may break contracts when proposing in a CC, and according to the definition of DACC, doctors do not receive compensation even if this constitutes a deception. To fill this gap, I prove the following lemma.

Lemma 4. *When contracts are substitutes for hospitals, there exists a number M such that after round M, no hospital ever deceives a doctor during a CC.*

Proving the lemma will finish the proof of Theorem 2; the same arguments from Claim 3 can be then used to show that $\boldsymbol{d}_i$ converges to zero for every doctor.

Proof of Lemma 4. Take M to be a round in which $\boldsymbol{d}_i$ is equal to zero for all hospitals and will remain zero at the end of each subsequent round[20]. In any round after M, if a hospital h is proposing some contract x in a CC, it must be that it has just lost some other contract x'. Prior to that, $x' \in Y_h = C_h(B_h)$ because $\boldsymbol{d}_h$ was equal to zero. Towards a contradiction, suppose that some doctor d is deceived because h breaks contract $y \in Y_h$ with d when the new contract x is signed. This would mean that

$$y \in R_h(Y_h \setminus \{x'\} \cup \{x\})$$

but since $Y_h \setminus \{x'\} \cup \{x\} \subset B_h$, substitutability of h's preferences implies that $y \in R_h(B_h)$, contradicting $y \in Y_h = C_h(B_h)$. □

[20] Such a round M exists by the arguments used in the proof of Claim 3. Note that the statement emphasizes the **end** of each subsequent round. In any round after M, a hospital may lose a contract (which leads to a jump in its $\boldsymbol{d}_i$) but a CC will bring $\boldsymbol{d}_i$ back to zero before the round is finished.

D.4 Generating all Stable Allocations

One could expect the second part of Theorem 1 to generalize to matching with contracts as easily as the first the part. However, this is not the case for a trivial reason.

Example 4. Suppose there is one hospital and one doctor. Let $X = \{l, m, h, \emptyset\}$. Suppose the doctor prefers h to m to l, and the hospital has a reversed ranking (think of the contract specifying the salary of the doctor). Then, there are three stable allocations, and the stable allocation $\{m\}$ cannot be achieved by DACC.

The problem of not being able to reach $\{m\}$ in the above example could be solved by allowing agents to break contracts *before* they propose a new one. But then, the extreme stable allocations l and h could not be achieved. To avoid this problem, I maintain the following assumption throughout.

Assumption 1. The set X and agents' preferences are such that for any two stable allocations $Y^\star$, $Z^\star$, for any $y \in Y^\star$, $z \in Z^\star$, if $d_y = d_z$ and $h_y = h_z$, then $y = z$.

Assumption 1 states that if the same agents sign a contract in two different stable allocations, then they sign the same contract. Under this assumption, if contracts are substitutes, the question whether the DACC can achieve all stable allocations is open once more. I found myself unable to provide a definitive answer. The proof method proposed in Section 3 does not work for general substitutable preferences. However, an adjusted method works under stronger assumptions on preferences, giving a partial converse to Theorem 2. This is the subject of the rest of this section.

I first define concepts that will be used to state the result.

Definition 6 (Dominance). Given two sets of contracts $Y^\star$, $Y \subset X$, I say that $Y^\star$ *dominates* Y for agent i if

1. $\forall y \in Y^\star \setminus Y,\ \ y \in C_i(Y \cup \{y\})$,

2. $\forall y \in Y^\star \cap Y, \forall z \in Y^\star,\ \ y \in C_i(Y \cup \{z\})$.

I say that $Y^\star$ *strongly dominates* Y if property 2 holds for $z \in X$, not just $z \in Y^\star$.

Although it is convenient to define dominance for arbitrary sets of contracts, it really has bite when the sets are acceptable. The following observation follows directly from the definition (by using SIRC).

Observation 5. *$Y^\star$ dominates Y for agent i if and only if $Y^\star$ is acceptable and $Y^\star$ dominates $C_i(Y)$.*

If $Y^\star$ dominates Y, then *(1)* every contract from $Y^\star$ that is not in Y is accepted by an agent holding contracts Y, and *(2)* if a new contract from $Y^\star$ is signed when the agent holds Y, then no contract from $Y^\star$ will be broken.

Dominance will play a key role in the proof. The idea, analogous to that used in the one-to-one setting in Section 3, is to make sure that the stable allocation we are trying to achieve dominates the current allocation for (almost) every agent. This means that no stable contract will be rejected or broken in the process. The next definition formulates two properties needed for the proof technique to work.

Definition 7 (Revealed Preference and Free-slot). Suppose that $Y^\star$ is acceptable, $Y^\star$ dominates Y, but $Y^\star$ does not dominate $C_i(Y \cup \{x\})$ for some contract $x \in X_i$. I say that preferences of agent i satisfy:

1. the *revealed preference* (RP) property, if for all such $Y^\star$, Y, x,

$$z \in C_i(Y \cup \{x\}) \implies z \in C_i(Y^\star \cup \{z\});$$

2. the *free-slot* (FS) property, if for all such $Y^\star$, Y, x,

$$z \in C_i(Y \cup \{x\}) \implies Y^\star \text{ dominates } C_i(Y \cup \{x\}) \setminus \{z\},$$

$$z \in C_i(Y),\, x \notin Y^\star \implies \forall y \in Y^\star \cap Y,\, y \in C_i(Y \cup \{x\} \setminus \{z\}).$$

The interpretation of these properties and their relation to other properties is discussed in detail in the next subsection.

Theorem 3. *Suppose that preferences of hospitals satisfy substitutes, RP, and FS properties. For an arbitrary stable allocation $Y^\star$, there exists a sequence Φ such that $Y^\star$ is the outcome of DACC(Φ). Moreover, $Y^\star$ can be achieved with an order Φ that does not lead to any compensation chains.*

The theorem is proved in Subsection D.4.2.

D.4.1 Discussion of Assumptions

The RP and FS properties as stated in Definition 7 are fairly complicated. Rather than trying to give their economic interpretation, I define stronger but more intuitive counterparts which imply them (the proof is by direct inspection and thus skipped).

Observation 6 (Simpler sufficient conditions). *Suppose that $Y^\star$ is acceptable, and Y is an arbitrary set of contracts. Preferences of agent i satisfy:*

1. *the strong RP property, if, whenever* $Y^\star$ *does not dominate* Y*,*

$$z \in C_i(Y) \implies z \in C_i(Y^\star \cup \{z\});$$

2. *the strong FS property, if, whenever* $Y^\star$ *dominates* $Y \setminus \{x\}$ *for some* x*,*

$$z \in C_i(Y) \implies Y^\star \text{ strongly dominates } C_i(Y) \setminus \{z\}.$$

The strong RP (FS) property implies the RP (FS) property.

The strong RP property can be interpreted as follows. If $Y^\star$ does not dominate Y, it means intuitively that some contract in $Y^\star$ is less attractive than contracts chosen from Y. The strong RP property says that in this case any contract chosen from Y should be accepted by an agent holding $Y^\star$ (presumably because it replaces the less attractive contract in $Y^\star$). As for the strong FS property, suppose that $Y^\star$ dominates Y once we remove one contract from it. Then, if any contract that is chosen from Y is removed, the resulting smaller set should be (strongly) dominated by $Y^\star$. The informal idea here is that the slot created by removing a signed contract could be filled with any contract from $Y^\star$. Both properties are satisfied when preferences are responsive.

Definition 8. Preferences of agent i are responsive if there exists a total order $\succ^i$ on X_i, and a natural number k_i such that

$$C_i(Y) = \{x \in Y : x \succ^i \emptyset, \, |\{y \in Y : y \succ^i x\}| < k_i\}, \, \forall Y \subset X_i. \tag{D.2}$$

Intuitively, agent i accepts the best contracts according to $\succ^i$ up to capacity k_i. The notion of dominance takes a straightforward form under responsive preferences. An acceptable $Y^\star$ dominates Y if and only if either $|C_i(Y)| < k_i$ or $|C_i(Y)| = k_i$ and the least preferred contract in $Y^\star$ is preferred to the least preferred contract in $C_i(Y)$.

Lemma 5. *If preferences of agent* i *are responsive, then they satisfy the strong RP and FS properties.*

Proof. RP: If Y is not dominated by $Y^\star$, then either there exists $y \in Y^\star \setminus Y$ such that $y \notin C_i(Y \cup \{y\})$, or there exists $y \in Y^\star \cap Y$ and $x \in Y^\star$ such that $y \notin C_i(Y \cup \{x\})$. In both cases, when $z \in C_i(Y)$, we must have $z \succ^i y$, and hence $z \in C_i(Y^\star \cup \{z\})$.

FS: This property follows immediately from the fact that under responsive preferences every acceptable $Y^\star$ strongly dominates any Y with $|Y| < k_i$. □

Preferences satisfying the RP and FS properties neither imply, nor are implied by substitutable preferences. Suppose hospital h has access to two doctors, and the only acceptable sets of contracts are to employ both or none. These preferences trivially satisfy Definition 7 but are not substitutable. On the other hand, consider preferences induced by the following ranking over 4 doctors:

$$\{d_1, d_2\} \succ^h \{d_1, d_4\} \succ^h \{d_2, d_3\} \succ^h \{d_2, d_4\} \succ^h \{d_3, d_4\} \succ^h d_1 \succ^h d_2 \succ^h d_3 \succ^h d_4.$$

These preferences satisfy substitutes. They satisfy neither the RP nor the FS property. Suppose that the hospital signs $Y_h^\star = \{d_2, d_3\}$ in the stable allocation. Let $Y = \{d_3, d_4\}$ be the set of contracts h temporarily signs in the course of the algorithm. $Y_h^\star$ dominates Y. However, suppose that h proposes a contract to d_1, and d_1 accepts. Then, h will break the stable contract d_3. This poses a problem for the proof which relies heavily on the property that stable contracts are never rejected or broken.

The RP property is violated in this example because $z = d_4$ is chosen from $Y \cup \{x\} = \{d_1, d_3, d_4\}$, but rejected from $Y^\star \cup \{z\} = \{d_2, d_3, d_4\}$. Intuitively, d_1 and d_4 exhibit indirect complementarity: d_4 is less attractive than the stable d_3 on its own but becomes more attractive in combination with d_1.

The FS property is violated because $z = d_4$ is chosen from $Y \cup \{x\} = \{d_1, d_3, d_4\}$, but $Y^\star = \{d_2, d_3\}$ does not dominate $C_h(Y \cup \{x\}) \setminus \{z\} = \{d_1, d_4\} \setminus \{d_4\} = \{d_1\}$. The issue here is that $\{d_1, d_3\}$ is not acceptable; even though a "free slot" is created by removing d_4, d_3 is not accepted when d_1 is employed.

D.4.2 Proof of Theorem 3

The idea of the proof of Claim 4 from Section 3 is to immediately break pairs that are not stable, in order to guarantee that no stable match partner will be rejected. Because hospitals can sign multiple contracts, we cannot guarantee that they will immediately dispose of unstable contracts when they are allowed to propose. Moreover, stable contracts can be broken when hospitals apply, an additional event we want to avoid.

In the proof, I make sure that there is at most one agent that could reject or break stable contracts, i.e. for whom the stable allocation does not dominate the current allocation. I say that agent i is *red-flagged* if $Y_i^\star$ does not dominate i's current allocation Y_i.

Fix a stable allocation $Y^\star$. I construct the sequence Φ recursively. Suppose that $k \geq 0$ proposers were already chosen. Consider the following properties:

(a) At the end of round k, at most one agent i is red-flagged. For that agent, $Y_i^\star$ dominates $Y_i \setminus \{z\}$ for some contract $z \notin Y_i^\star$.

(b) Up to (and including) round k, there haven't been any CCs.

(c) Up to (and including) round k, no contract has been added to B_i, for all i.

(d) Up to (and including) round k, $Y_i^\star \subset B_i$, for all i.

The first element $\Phi(1)$ is an arbitrary doctor. In round $k+1$, if the DACC algorithm has not stopped, I choose $\Phi(k+1)$ as a function of the outcome of round k.

Case 1) If there is a red-flagged agent i, choose $\Phi(k+1) = i_z^c$ for some contract $z \notin Y_i^\star$ such that $Y_i^\star$ dominates $Y_i \setminus \{z\}$;

Case 2) otherwise, set $\Phi(k+1)$ to be an arbitrary doctor d with $Y_d \neq Y_d^\star$; if no such doctor exists, choose a hospital h with $Y_h \neq Y_h^\star$. (If no such hospital exists, the algorithm has stopped.)

If properties (a)-(d) hold at k, the choice of $\Phi(k+1)$ is well defined because there is at most one red-flagged agent i, and the contract z can be found for i. These properties hold trivially for $k = 0$ (before the algorithm starts). If they hold for all k until the algorithm stops at K, then we are done. By the stopping criterion, $C_i(B_i) = Y_i$ for each agent i. For all j for whom $Y_j^\star$ dominates Y_j, we have $Y_j = Y_j^\star$, as otherwise $\boldsymbol{d}_j(B_j, Y_j) > 0$, contradicting Lemma 3.[21] By property (a), there is at most one agent i for whom $Y_i^\star$ does not dominate Y_i. But because Y_i is acceptable, and contracts are bilateral, $\{Y_j\}_{j \neq i}$ uniquely determines Y_i, and thus $Y_i = Y_i^\star$ as well.

I now assume that properties (a) - (d) hold at k (the inductive hypothesis), choose the proposer $\Phi(k+1)$ accordingly, and prove that properties (a) - (d) hold at $k+1$ (the inductive step). I start with a simple lemma that will be used throughout.

Lemma 6. *Up to round k, no contract x was proposed by both its counterparties j and j_x^c. When j proposes x, contract x is already in j_x^c's budget set.*

Proof. Suppose that there is a contract x, with sides j and $l = j_x^c$, and both j and l proposed contract x at some point. By the inductive hypothesis, property (d), when contract x was proposed, the stable allocation was in the proposer's budget set. By substitutes, when x is chosen from a set, it is also chosen from any subset thereof. This means that $x \in C_j(Y_j^\star \cup \{x\})$ and $x \in C_l(Y_l^\star \cup \{x\})$. Stability of $Y^\star$ requires $x \in Y^\star$. But in this case x could not be proposed twice because stable contracts are always accepted up to round k, by property (d).

Further, suppose that j proposes x. By the above, we know that j_x^c did not propose x. By the inductive hypothesis, there were no CCs up to round k, so j_x^c could not be deceived. Thus, x must be in j_x^c's budget set when x is proposed. $\square$

The lemma can be used to prove a further useful fact.

[21] Strictly speaking, Lemma 3 only implies that $Y_j^\star \subseteq Y_j$ but a strict inclusion can easily be shown to violate stability of $Y^\star$.

Lemma 7. *If the inductive hypothesis holds up to round k, there are no deceptions in round $k+1$.*

Proof. I consider two cases: when an agent proposes, and when she receives an offer.

Suppose that i with allocation Y_i proposes a contract x, signs x, and breaks some contract $y \in Y_i$, deceiving i^c_y. This means that i proposed y before. By the definition of DACC, $Y_i \cup \{x\}$ is in i's budget set. By property (c) of the inductive hypothesis, budgets can only decrease, so $Y_i \cup \{x\}$ was in i's budget set when y was proposed. By substitutes, y needs to be chosen from $Y_i \cup \{x\}$, as it was chosen from a larger set containing $Y_i \cup \{x\}$. This is a contradiction.

Now suppose that agent i receives an offer x, accepts it, and breaks some contract $y \in Y_i$, deceiving i^c_y. By Lemma 6, x was already in i's budget set when it was proposed. The rest of the proof is the same as in the previous case. $\square$

The inductive step Consider round $k+1$. Property (b) holds in round $k+1$ by Lemma 7. Because there are no CCs, there is only one proposed contract in round $k+1$. By the second part of Lemma 6, the proposed contract is already in the receiving agent's budget set, which establishes property (c). I prove properties (a) and (d) below.

If the agent proposing in round $k+1$ is rejected, property (a) is obvious (no new agent can become red-flagged). To show property (d), it is enough to prove that the agent receiving the offer would not reject a contract that is included in the stable allocation. The inductive hypothesis, property (a), and the way we chose $\Phi(k+1)$, imply that the receiving agent is not red-flagged, and the conclusion follows directly from the definition of dominance (Definition 6).[22]

From now on, I focus on the case when the offer in round $k+1$ is accepted. I will consider two cases, depending on how the proposer was determined by the outcome of the previous round.

Case 1 in round k In this case, by the inductive hypothesis and the way we chose Φ, the situation can be summarized as follows. Agent i is red-flagged, but $Y_i^\star$ dominates $Y_i \setminus \{z\}$ for some contract $z \notin Y_i^\star$. No other agent is red-flagged. The current proposer is $j = i^c_z$, the counterparty of i under contract z. Agent j proposes some contract x to $l = j^c_x$, and this offer is accepted.

To establish property (a) for round $k+1$, I prove three lemmas.

Lemma 8. *Suppose agent j holds an acceptable set of contracts Y_j which is dominated by $Y_j^\star$. Suppose j signs contract x, and the new set of contracts $Y_j' = C_j(Y_j \cup \{x\})$*

[22] The only exception is when j proposes back to i. This is possible because i and j can in general sign different contracts with each other. In this case, we use the definition of stability and Assumption 1

is no longer dominated by $Y_j^\star$ (j becomes red-flagged). Then, there exists a contract $y \in Y_j' \setminus Y_j^\star$ such that $Y_j^\star$ dominates $Y_j' \setminus \{y\}$.

Proof. If $x \notin Y_j^\star$, set $y = x$. Suppose that $x \in Y_j^\star$. Because $Y_j^\star$ does not dominate Y_j', we have $Y_j' \nsubseteq Y_j^\star$, and so there exists $y \in Y_j' \setminus Y_j^\star$. The set Y_j' is acceptable, so contract y is chosen from Y_j'. By the FS property (first implication), $Y_j^\star$ dominates $Y_j' \setminus \{y\}$. □

Lemma 8 provides a way to choose a contract that will satisfy property (a) for the red-flagged agent (y from Lemma 8 becomes the contract z in property (a) for the next round). So I only need to prove that at most one agent is red-flagged.

Lemma 9. *If j becomes red-flagged, then i stops being red-flagged.*

Proof. I prove that if j becomes red-flagged, then j must break contract z with i which means that i will no longer be red-flagged (by property (a) for round k).

Suppose otherwise, i.e. z is still chosen after j signs the contract x with l, $z \in C_j(Y_j \cup \{x\})$. Since j becomes red-flagged, $Y_j^\star$ does not dominate j's current allocation $C_j(Y_j \cup \{x\})$. By the RP property, $z \in C_j(Y_j^\star \cup \{z\})$. Similarly for i, we know that $z \in C_i(Y_i)$, and because i was red-flagged, $Y_i^\star$ does not dominate Y_i. Hence, $z \in C_i(Y_i^\star \cup \{z\})$. This contradicts stability of $Y^\star$; contract $z \notin Y^\star$ would be signed by i and j at $Y^\star$. □

Lemma 10. *When $x \notin Y^\star$ and $l = j_x^c$ accepts the contract x proposed by j, l doesn't become red-flagged.*

Proof. Suppose that $x \notin Y^\star$. Because $Y_j^\star \subset B_j$, and $x \in C_j(B_j)$, by substitutes, x needs to be chosen from $Y_j^\star$ as well, so we have $x \in C_j(Y_j^\star \cup \{x\})$. Suppose that l becomes red-flagged, i.e. $Y_l^\star$ no longer dominates $C_l(Y_l \cup \{x\})$. Because $x \in C_l(Y_l \cup \{x\})$, by the RP property, $x \in C_l(Y_l^\star \cup \{x\})$. This contradicts stability of $Y^\star$. □

I now finish the proof of property (a) by showing that at most one of agents i, j, or l is red-flagged. If $x \notin Y^\star$, then the conclusion follows directly from Lemma 9 and 10. If $x \in Y^\star$, then there are two cases: either j is a doctor or a hospital. If j is a doctor, then j breaks z when she signs x, so agent i is not red-flagged. Because j is a doctor signing her stable contract x, j cannot become red-flagged. So in this case only l can be red-flagged. If j is a hospital, then l is a doctor so l does not become red-flagged. From Lemma 9, if j becomes red-flagged, then i stops being red-flagged, so at most one of them can be red-flagged.

To prove property (d), I have to show that (i) j does not break some contract included in $Y_j^\star$ when contract x is signed, and (ii) l does not break some contract included in $Y_l^\star$ when she accepts x.

As for (i): If j is not red-flagged at the end of the round, then, by definition, j could not break any contracts from the stable allocation. If j becomes red-flagged, then (by Lemma 9) j must have broken the contract z with i when j signed x. Contract z is chosen from Y_j, j's initial allocation (before signing x) which is dominated by $Y_j^\star$. If $x \notin Y_j^\star$, we can use the second implication in the FS property, obtaining that no stable contract is discarded from $C_j(Y_j \cup \{x\} \setminus \{z\})$. Because $z \notin C_j(Y_j \cup \{x\})$, by SIRC, $C_j(Y_j \cup \{x\} \setminus \{z\}) = C_j(Y_j \cup \{x\})$, and thus the conclusion holds in this case. If $x \in Y_j^\star$, then j could not break any stable contract by the fact that j was not red-flagged prior to signing x (this follows from the definition of dominance).

As for (ii): If l is not red-flagged at the end of the round, then l could not break any contracts from the stable allocation. If l is red-flagged, then we know from the above analysis that $x \in Y_l^\star$ and l is a hospital. Because l was not red-flagged prior to accepting x, the conclusion follows from the definition of dominance.

Case 2 in round k In this case, by the inductive hypothesis and the way we chose Φ, the situation can be summarized as follows. No agent is red-flagged. The current proposer is j who is an agent with $Y_j \neq Y_j^\star$. Agent j proposes a contract $x \in C_j(B_j) \setminus Y_j$.

If j is a hospital, then j's offer is rejected. This follows from the fact that a hospital is chosen as the proposer under Case 2 only if $Y_d = Y_d^\star$ for all $d \in D$. Acceptance of j's offer would violate stability of $Y^\star$, so x is rejected. The case of rejection of x has been already covered.

From now on, I can assume that j is a doctor and $l = j_x^c$ accepts x.

Consider property (a). Lemma 8 still applies, so I only need to prove that at most one agent becomes red-flagged (since there are currently no red-flagged agents). This follows directly from Lemma 10 if $x \notin Y^\star$. And if $x \in Y^\star$, because j is a doctor, j cannot become red-flagged.

Consider property (d). I have to show that (i) j does not break some contract included in $Y_j^\star$ when contract x is accepted, and (ii) l does not break some contract included in $Y_l^\star$ when she accepts x. As for (i), this cannot happen because, by the way we choose Φ, j would not be chosen as the proposer if j's current allocation were stable. As for (ii), we can use the same argument as in Case 1.

This finishes the inductive step, and hence the proof of the theorem.