Deferred Acceptance with Compensation Chains

Piotr Dworczak[*]

August 14, 2019

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. The proof of convergence of DACC algorithms uses a novel technique based on a construction of a potential function.

Subject Classifications: networks/graphs: matchings; games/group decisions: cooperative

[*] Department of Economics, Northwestern University, email: piotr.dworczak@northwestern.edu.

1 Introduction

Deferred Acceptance (DA) algorithms play a central role in matching theory. In a seminal paper, Gale and Shapley (1962) introduced 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 (see Roth, 2008a). 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 (2008a). A stable mechanism is advocated for cadet-branch matching in the US army by Sönmez and Switzer (2013).

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 more general sequences of proposers? Is the outcome always stable? Can all stable matchings be reached if we let agents from both sides of the market propose? While these questions seem quite fundamental, many issues remain open.

In this paper, I explore the connection between stability and a general class of deferred acceptance algorithms (DACC) to address some of these open questions. A Deferred Acceptance with Compensation Chains algorithm works as follows. Agents make offers one at a time according to a pre-specified (arbitrary) sequence. When proposing, agents make an offer to the best available partner; when receiving an offer, agents temporarily accept (hold) an offer if they prefer the proposer to the current match. Initially, all agents on the other side of the market are available. Partners become unavailable to agent i when they reject or divorce i, and become available when they propose to 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. When an agent is deceived, he or she gets *compensated*, i.e., is allowed to make an additional offer, out of turn.

Apart from allowing an arbitrary sequence of proposers, compensation is essentially the only difference relative to the Gale-Shapley algorithm. In particular, when all offers are made by one side of the market, a deception cannot take place, and the above procedure is equivalent to the Gale-Shapley algorithm.

The main result is that for any sequence of proposers, the DACC algorithm converges (in polynomial time) to a stable outcome. Conversely, for any stable outcome, there exists a sequence of proposers such that DACC converges to that stable outcome. Below, I discuss three reasons for the significance of this finding.

First, on the practical side, the DACC class can be an attractive matching algorithm for markets in which the designer is concerned about procedural fairness (see Klaus and Klijn, 2006, for an insightful discussion). Because DACC allows for an arbitrary sequence of proposers, there exist DACC algorithms that treat the two sides of the market symmetrically, for example, when the sequence of proposers is drawn uniformly at random. Compared to a fair randomization over the men- and women-proposing Gale-Shapley algorithm, DACC lowers the variance of outcomes (measured by the rank of the stable-match partner) because it produces non-extreme stable matchings with positive probability (the improvement is strict when there are more than two stable matchings). Other examples of fair stable mechanisms include Ma (1996), Romero-Medina (2005), Klaus and Klijn (2006), and Kuvalekar (2015). However, only DACC has the property that *all* stable matchings can be reached. For example, the random order mechanism (Ma, 1996), employment by lotto (Aldershof *et al.*, 1999, and Klaus and Klijn, 2006) or the greedy correcting procedure (Blum and Rothblum, 2002) cannot reach "strictly interior" stable matchings in which no agent gets her most preferred stable match partner. As a corollary, DACC will have a lower variance than these mechanisms in markets where the only stable matchings are either men-optimal, women-optimal, or strictly interior (see Klaus and Klijn, 2006, or Section 3 for examples of such markets).

Procedural fairness is complementary to other notions of fairness, such as the median matching (see Teo and Sethuraman, 1998 and Schwarz and Yenmez, 2011) which can be viewed as "outcome fairness." 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. Unlike other procedurally fair algorithms, the DACC algorithm with random order of proposers reaches the median matching with positive probability whenever it exists, and is thus a compromise between procedural and outcome fairness. (Importantly for applications, the DACC algorithm can be easily extended to many-to-one matching with contracts, and converges to a stable outcome under the usual substitutability condition; this result can be found in an earlier working version of this note.)

Second, the equivalence between deferred acceptance and stability is obtained using a novel proof technique for showing convergence. To the best of my knowledge, convergence of existing DA-like algorithms is based on monotonicity of the offer process; In DACC algorithms, neither the set of partners available to any agent, nor the match utility of any side changes monotonically. Instead, I construct a potential function which, for any agent, measures the distance (in the preference ordering) between that agent's current match and the most preferred partner that is available. The potential function decreases in the steps of the algorithm in which no agent is divorced. The inclusion of compensation chains in the DACC algorithm serves the sole purpose of guaranteeing convergence by ensuring that the potential function does not increase permanently following a deception, i.e., a withdrawal of a previously made offer. 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. Convergence of DACC is reminiscent of the Edgeworth process in which a barter economy achieves an equilibrium through local Pareto improving trades (see Uzawa, 1962, and more recently, Axtell, 2005) and, on a more abstract level, the tâtonnement process for prices in competitive equilibrium (see Arrow and Hurwicz, 1958, and Uzawa, 1960). 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.

Finally, the main result establishes an equivalence between (properly generalized) deferred acceptance procedures and stability: A matching is stable if and only if it is the outcome of a DACC algorithm. In particular, if stability is among the design goals in some market, there is no loss of generality in restricting attention to deferred acceptance algorithms. Previous papers showed a tight connection between these concepts but did not demonstrate equivalence. In the algorithms proposed by Ma (1996) (based on Roth and Vande Vate, 1990), Blum and Rothblum (2002), or Kesten (2004), both sides of the market make offers but not all stable matchings can be reached. In contrast, McVitie and Wilson (1971) propose a deferred acceptance algorithm that generates a superset of stable matchings – it may lead to non-stable matchings that must be manually discarded (see also Balinski and Ratier, 1998). While there exist methods for generating the set of stable matchings, these characterizations use non-deferred-acceptance algorithms relying on more complex mathematical objects: Irving and Leather (1986) use rotations, Adachi (2000), Fleiner (2003), Echenique and Oviedo (2004), and Hatfield and Milgrom (2005) – pre-matchings and fixed-point theory on lattices, Kuvalekar (2015) – graph theory. These algorithms are generally superior from a computational point of view and provide more insights about the structure of the core. The appeal of DACC lies in its relative simplicity, especially from the point of view of real-life agents. As is pointed out by Roth (2008b), it is often important in practical market design that agents understand the mechanism and find it easy to participate. DACC uses the same language as the original Gale-Shapley algorithm and is easy to describe in non-technical terms.

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, $\mu(i) = \operatorname{argmax}\left\{B_i; \succ^i\right\}$ (where $\operatorname{argmax}\{A; \succ\}$ is defined as the most preferred option from the set A with respect to the order $\succ$). 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 μ−partner. (A version of this condition was previously used by Adachi, 2000; see also Echenique and Oviedo, 2004.)

I conclude this section with two remarks about the Gale-Shapley algorithm (which I will sometimes refer to as One-Sided Deferred Acceptance, or 1DA). First, the order in which men propose in 1DA does not play any role. Instead of simultaneous proposals, men could propose one-by-one, and women could make (tentative) acceptance decisions by evaluating the proposer against their current match (see McVitie and Wilson, 1971, for an algorithm based on this observation). Second, 1DA can be described in the language of budget sets (see, for example, Hatfield and Milgrom, 2005). In the men-proposing version, each men starts with a budget set consisting of all women, and each woman starts with an empty budget set. In every round, each man proposes to the most preferred woman in his budget set. Proposers 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 sequence. 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. If i is already matched to j, or if there are no acceptable partners in i's budget set, we skip the round. 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, 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.

It is possible that two agents (from opposite sides) need to be compensated in the same round, and thus two compensation chains begin in the same round. It is irrelevant which chain is run first. However, when the chains cross, that is, when an agent i that is being compensated proposes to agent j who is waiting to be compensated, the order switches: j is compensated first, and only then i's compensation continues. I use this property to prove that both chains are guaranteed to stop.

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 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 proposes (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.

Before stating the main results, I illustrate how the DACC algorithm works using a simple example. Suppose that there are three men and three women, and preferences are given by:

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

There are three stable matchings: the men-optimal $\mu^M = \{\{m_1, w_1\}, \{m_2, w_2\}, \{m_3, w_3\}\}$, the 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 that the DACC algorithm can generate $\mu^\star$, consider the sequence

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

In each of the first six rounds, an agent proposes to their favorite partner, and subsequently gets divorced in the following round (when that partner becomes the next proposer). 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 $\mu^\star$. There are no deceptions, and hence no compensation chains.

To see how compensation works, assume that w_1 finds m_3 unacceptable (this simplifies the example), and consider the following sequence:

$$\Phi = w_1, m_1, m_1, m_2, m_2, w_1, \ldots .$$

Because w_1 is matched to m_2 in round 2, m_1 is rejected in round 2 and hence matches to w_2 in round 3. However, w_1 loses the match with m_2 because he proposes in round 5 to w_3 (after being rejected by w_2 in round 4). Then, in round 6, w_1 proposes back to m_1 (m_3 is assumed

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 proposed 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. $i$ proposes;
        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. If $j = \text{argmax}\{B_i; \succ^i\} \in CC$, then    (If two compensation chains cross...)
            A. Until $(\mu(j) = \emptyset$ and $B_j \neq \emptyset)$, then $j$ proposes;    (...compensate $j$ before $i$)
            B. Remove $j$ from $CC$;
        iii. $i$ proposes;
        iv. 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 proposes"

```
1. Set $j := \text{argmax}\{B_i; \succ^i\}$;   ($i$ proposes to $j$)
2. If $\{i, j\} \in \mu$ or $j = \emptyset$ then return; else:    (If $i$ and $j$ are already matched or $i$ proposes to $\emptyset$)
3. Set $A_j := A_j \cup \{i\}$ and $B_j := B_j \cup \{i\}$;   (Record that $i$ proposed 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.

unacceptable). Because m_1 prefers w_1 to w_2, he divorces w_2 although he proposed to her before. Thus, w_2 is deceived, and gets to propose before the sequence Φ is continued. She proposes to m_3, m_3 accepts, and we reach a stable matching.

Theorem 1. *For any sequence Φ, DACC(Φ) stops in polynomial 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 relabeling, either *(i)* i proposed 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 proposes 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 formally define a (single) CC. Consider an instance in the k-th round of the DACC algorithm when $\Phi(k)$ proposes and causes a divorce of some agent i by $j \in A_i$ (i.e. j deceives i). This could be either because $\Phi(k) = j$, i.e. i and $\Phi(k)$ were matched, or that $\Phi(k)$ proposed to j who was matched to i. Then we initiate a CC at i. Let $i_0 = i$. Fixing a sequence of agents $(i_0, i_1, ..., i_{m-1})$ who proposed in that CC so far, I show how to choose i_m. If i_{m-1} proposed and was rejected, choose $i_m = i_{m-1}$. If i_{m-1} proposed 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. There are at most two CCs in any round k.*

Proof. If only one CC is initiated in a round, or the two CCs never cross (no agent in a CC proposes to an agent waiting to be compensated in the other CC), then the claim follows from three observations. First, when the chains do not cross, the second chain is only run after the first one stops. Second, in a CC initiated at a man, only men propose (analogously for women). Third, 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 (analogously for women). Thus every single CC must stop eventually, and hence in any round there are at most two CCs.

Next, consider the case when two CCs cross. It is enough to prove that the first chain (without loss, a chain for men) stops without triggering a new chain for the same side (men). When the chains cross – a man m that is being compensated is about to propose to a woman w_0 that awaits compensation – w_0 is compensated first and there are three cases. First, w_0 may propose back to m (possibly after getting rejected by some other men first). In this case, both chains stop immediately because both m and w_0 get matched to the best options in their respective budget sets. Second, w_0 may be compensated without causing any deception. In this case, the women's CC stops, and thus the men's CC stops for the reasons described in the first part of the proof (since it's the only CC remaining). Finally, suppose that w_0 gets compensated and some other w_1 gets deceived as a result (and hence awaits compensation). In this case, m is compensated before w_1 and will be rejected by w_0 before making the next offer. If m is compensated in any way other than by proposing to w_1, then the men's CC will stop. If m ever wants to propose to w_1 as part of his compensation, then we can repeat the above reasoning with w_0 replaced with w_1: Thus, either the chains stop, or m will be rejected by w_1 and will propose to some w_2 that awaits to be compensated. Note that m prefers w_0 to w_1, w_1 to w_2, etc. Therefore, eventually, because m has a finite preference list, m must either be left with an empty budget set or be accepted by a woman who does not await to be compensated. In either case, the chain for men stops. □

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 $i \in N$:

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

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. The next lemma shows that d is a potential function.

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 relegated to Appendix A. I sketch it below. By direct inspection, the function d_i decreases weakly when agent i receives an offer (except when that agent waits to be compensated), and decreases strictly when agent i proposes. 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 proposed 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 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 converges in finite time to a stable matching.*

Observation 2 constitutes an alternative proof of the main result of Roth and Vande Vate (1990).

The above proof technique, based on the potential function, can be used to show that DACC requires a polynomial (in the number n of agents) number of non-trivial proposals to converge. (By a non-trivial proposal I mean a proposal that is made to a partner with whom the proposer is not already matched.) Every compensation chain requires at most n^2 steps to stop (because only one side of the market proposes in any CC). As shown in the proof of Lemma 1, the potential function is strictly decreasing, except for an $O(n^2)$ number of rounds, and rounds in which there are trivial proposals. Therefore, we obtain the following result.

Corollary 1. *The DACC algorithm requires at most $O(n^4)$ non-trivial proposals to converge.*

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 $\Phi(k)$ proposed 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 proposes. 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 proposed to someone less preferred to $\mu(i)$. In particular, in round $k+1$, i proposes 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 proposed to i in round k. Thus, by property (c), l prefers i to $\mu(l)$. If i proposed 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. □

4 Relationship to other DA algorithms

The key distinction between DACC and existing deferred acceptance algorithms is that DACC imposes minimal restrictions on both the sequence of proposers and the set of agents to whom a proposer may make an offer. This flexibility is responsible for DACC's property of reaching all stable matchings. At the same time, the unrestrictedness of the offer process makes the convergence of the algorithm more challenging. A contribution of this paper is to emphasize the role of compensation chains in ensuring convergence when the sequence of proposers is otherwise unrestricted – using a novel proof technique based on a potential function. The idea of compensation itself (understood as a divorced agent becoming the next proposer) is not new: its antecedents may be found in Roth and Vande Vate (1990), Ma (1996), Blum and Rothblum (2002), or Kojima and Ünver (2008).

I will discuss the detailed relationship using the *random order (RO) mechanism* of Ma (1996) as a representative example. (This algorithm is based on Roth and Vande Vate, 1990; a similar discussion applies to closely related algorithms in Blum *et al.*, 1997, Blum and Rothblum, 2002, and Klaus and Klijn, 2006.) The RO mechanism orders agents in a queue and then consecutively finds a stable matching on a growing set of agents by adding them one by one to the market. When a new agents arrives and causes a divorce, the divorced agent makes the next offer – similarly as in a compensation chain. There are three restrictions on the offer process relative to DACC:

1. Agents may only propose to other agents already in the market;
2. Only unmatched agents may propose;
3. Every divorce triggers a compensation.

The proof of Claim 4 - establishing that any stable matching can be reached – relies on the fact that DACC relaxes these restrictions:

1. Agents can propose to anyone in their budget set; fixing a target stable matching, when an agent proposes in DACC, he or she can in particular propose to her stable match partner (in Ma's algorithm, the stable match partner may not yet be in the market).
2. Matched agents can propose; the proof relies on breaking certain undesirable pairs by letting matched agents propose (even if the outcome among agents that proposed so far is stable).
3. Only a deception (which is a special type of divorce) triggers a compensation; Divorces cannot be avoided even in a one-sided deferred acceptance algorithm, and thus running a compensation after every divorce is restrictive. Deceptions, on the other hand, are much rarer events, and, as Claim 4 demonstrates, can be avoided on the way to any stable matching.

Compensation chains are needed to guarantee convergence of DACC for any sequence of proposers but are not required for other properties. To show this, in Appendix B, I introduce two simplifications of DACC. First, I define an algorithm that is identical except that it does not include compensation chains; then, all properties hold apart from convergence – I construct a matching market and a sequence of proposers such that the algorithm falls into an infinite loop. (However, if the algorithm stops, it produces a stable matching.) Second, to restore convergence in the above algorithm, one could specify that budget sets can only decrease, that is, agents are not added back to the budget set of a partner to whom they propose. However, in this case, there exist sequences of proposers that lead to a non-stable outcome.

5 Concluding Remarks

This paper studies a one-to-one matching market to simplify exposition. However, because the definition of DACC does not rely on the two-sidedness of the marriage market, it can be applied (without any modifications) to the roommates problem. Under the "no-odd-rings" condition (see Chung, 2000), the results of the note generalize to this setting: Every sequence of proposers leads to a stable outcome, and any stable outcome can be reached by some sequence. A previous version of this paper extends the conclusions to a many-to-one matching under appropriate substitutability conditions. It would be interesting to see if DACC could work equally well in other settings, e.g., in many-to-many matching or the coalition formation problem (see Pycia, 2012).

The DACC algorithm is not in general strategy-proof for either side of the market. Strategy-proofness of a stable matching algorithm depends solely on which stable matching it eventually selects. The results of Sönmez (1999) imply 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 the sequence of proposers and the resulting stable matchings – a task left for future research.

Putting aside the issue of incentives, DACC could serve as a model for decentralized matching markets. The Gale-Shapley algorithm imposes rigid assumptions on the set of proposers and thus may fail to approximate the behavior of agents in many decentralized matching markets. DACC assumes much less about the structure of the offer process. The role of compensation chains in the proof of convergence loosely suggests that a decentralized market well-approximated by the DACC procedure will tend to stabilize if agents who are deceived are able to make new offers quickly.

Acknowledgments

I would like to thank Nick Arnosti, Peter Biro, Yi Chen, Fuhito Kojima, Rida Laraki, Shengwu Li, Giorgio Martini, Paul Milgrom, Alexandru Nichifor, Afshin Nikzad, Michael Ostrovsky, Al Roth, Andrzej Skrzypacz, Bumin Yenmez, Anthony Lee Zhang, and participants of several conferences and seminars for helpful comments and discussions.

I am deeply thankful to Panagiotis Karras and Nikos Tziavelis for pointing out a problem in the proof of Claim 2 and for subsequent help in fixing it. Converegece of DACC could also be obtained by modifying the acceptance rule within the CC along the lines of the Strongly Deferred Acceptance algorithm considered by Tziavelis (2018).

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.

Aldershof, B., Carducci, O. and C. Lorenc, D. (1999). Refined inequalities for stable marriage. *Constraints*, **4**, 281–292.

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

Axtell, R. (2005). The complexity of exchange*. *The Economic Journal*, **115** (504), F193–F210.

Balinski, M. and Ratier, G. (1998). Graphs and marriages. *The American Mathematical Monthly*, **105** (5), 430–445.

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.

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. (2004). Core many-to-one matchings by fixed-point methods. *Journal of Economic Theory*, **115**, 358–376.

Fernandez, M. A. (2017). Deferred Acceptance and Regret-Free Truth-telling: A Characterization Result, unpublished Manuscript.

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).

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

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

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.

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.

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

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. (2008a). Deferred acceptance algorithms: history, theory, practice, and open questions. *International Journal of Game Theory*, **36** (3), 537–569.

— (2008b). What have we learned from market design?*. *The Economic Journal*, **118** (527), 285–310.

— 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.

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

Tziavelis, N. (2018). *Algorithmoi gia dikaia kai stathera tairiasmata*. Master's thesis, School of Electrical and Computer Engineering, National Technical University of Athens, available at http://artemis.cslab.ece.ntua.gr:8080/jspui/handle/123456789/17048.

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

— (1962). On the stability of edgeworth's barter process. *International Economic Review*, **3** (2), 218–232.

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 who proposed 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 $d_{\Phi(k)}^{k-1} = 0$, 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 $d_{\Phi(k)}^{k-1} > 0$ 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 $l' \notin A_l^k$ 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, $d_i^k + d_j^k$ 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 propose in a CC end up being matched to the most preferred option in their respective budget sets. Thus, $d_m^k = 0$ for all m who propose 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 $l' \notin A_l^k$, i.e. we have a divorce which is not followed by a CC. Because l is divorced, we have $l' \notin B_l^k$. Because $l' \notin A_l^k$ and $A_l^k = A_l^{k+n}$ for any $n \in \mathbb{N}$ (because $k > k^\star$), $l' \notin B_l^{k+n}$ for all $n \in \mathbb{N}$. That is, l can never propose to l'. And due to $l' \notin A_l^{k+n}$ for all n, l' never proposes to l either. Thus, we add $\{l, l'\}$ to X^k, and thus $|X^k|$ grows strictly. □

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 $d_{\Phi(k)}^{k-1} = 0$ (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

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

The number a_n is well defined. Indeed, because the algorithm didn't stop at a_{n-1}, there exists an agent i with $d_i^{a_{n-1}} > 0$. 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 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 (Gale-Shapley algorithm) 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.

B.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. An example was provided in Section 3.

B.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 propose, i proposes 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 below. 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.

Example 1. (2DA may converge to a non-stable outcome.) Consider the following preferences:

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

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	$\Phi(k)$	proposes to	Decision	Current matches
1	w_1	m_2	accept	w_1m_2
2	m_2	w_2	accept	~~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	~~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	~~m_1w_3~~, ~~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	~~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 proposes to m_1 in round 4, m_1 is temporarily matched to a more preferred w_3. By the time when m_1 proposes 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 proposed 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 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 proposes 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.

Example 2. (B2DA may never converge.) Consider the following preferences:

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

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	$\Phi(k)$	proposes 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, ~~m_2w_1~~, m_3w_1
6 (+6n)	w_3	m_1	$w_3 \in B_{m_1}$, $m_1 \notin B_{w_2}$	~~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, ~~w_3m_1~~, m_2w_3
8 (+6n)	w_2	m_3	$w_2 \in B_{m_3}$, $m_3 \notin B_{w_1}$	~~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, ~~w_2m_3~~, m_1w_2
10 (+6n)	w_1	m_2	$w_1 \in B_{m_2}$, $m_2 \notin B_{w_3}$	~~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 a proposal from the most preferred partner j, and thus rejects the current partner. But then j receives a proposal 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 proposers. 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 proposed 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 proposes to w_2. Woman w_2 accepts the offer and a stable matching is reached.