UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FACULTY OF BUSINESS ADMINISTRATION AND ECONOMICS

Working Paper Series

Working Paper No. 2020-04

Stability in Weighted College Admissions Problems

Britta Hoyer and Nadja Stroh-Maraun

May 2020

Stability in Weighted College Admissions Problems

Britta Hoyer and Nadja Stroh-Maraun

Abstract

There are a number of college admissions problems in which students are heterogeneous according to the space they occupy at the college they are allocated to. To deal with this source of heterogeneity we propose a weighted college admissions problem by assigning each student a so-called weight. The existence of stable matchings is not ensured in weighted college admissions problems. To find a stable matching, if it exists, we propose a new algorithm, the deferred acceptance algorithm with gaps (DAG). It results in stable matchings if existing and cylces otherwise. Moreover, we show how to restore stability.

Keywords: Matching, School Choice, College Admissions Problems, Deferred Acceptance Algorithm, Stability, DAG, Gaps

JEL: C78, D47

1 Introduction

Many-to-one matching problems in two-sided markets (the so-called *college admissions problems*), have received a lot of attention in the matching literature throughout the years. They model situations in which students have preferences over colleges and colleges have preferences over students; while each student needs a place at exactly one college, each college has a certain quota denoting the maximal number of students it can accept. Next to the standard college admissions problems, in reality we observe a number of many-to-one matching situations in which some of the assumptions of the college admissions problem are not met. One of these situations takes place when students are not homogenous in terms of how much room they occupy at the institution they apply to. In other words, we have a matching problem with differently weighted participants. For example, across Japan (Kamada & Kojima, 2019) and Europe children are allocated to daycare facilities with the help of various, mostly decentralized matching mechanisms. The allocation, however, is constrained by regulations on staff-to-child ratios in daycare centers and kindergartens, usually differing with the age of the children.[1] Another setting in which matching mechanisms are similarly constrained are inclusive schools, where students with and without (learning) disabilities are taught jointly; however, different staff-to-student ratios are needed for each group. Finally, when matching students to supervisors for final theses, weights might play a role, as supervising a Master's thesis takes more time and effort than supervising a Bachelor thesis (Hoyer & Stroh-Maraun, 2020).

In each of these examples, we observe what we term a *weighted college admissions problem*. In this paper we analyze whether stable matchings exist in such a weighted college admissions problem, whether standard matching algorithms work well in this setting and how stable outcomes, if they exist, may

[1] The regulation for the kindergarten year 2012/2013 states that in Germany,for example, in the state of Rhineland-Palatinate, the staff-to-child ratio is one staff member for maximally five children for children under the age of three, whereas for children between three and six the ratio is one staff member for maximally 14 children (European Commission/EACEA/Eurydice/Eurostat, 2014).

be found. Building on previous research (see, e.g., McDermid & Manlove 2010; Delacrétaz 2019), it is easy to show that, unlike in the case of the standard college admissions problem, there does not always exist a stable matching in the setting with differently weighted students. Furthermore, we find that, even if it exists, the DA in a weighted college admissions problem does not necessarily lead to a stable outcome. To counteract this, we propose a new mechanism based on the DA by introducing the additional feature of *gaps*. We show that this "DA with gaps" (DAG) is able to determine whether a stable outcome exists and finds such a stable outcome if it exists. For the cases where no stable outcome exists, we show that we can guarantee the existence of a stable matching by increasing or decreasing some colleges' quotas, where the size of the adjustment is smaller than the highest weight of one student.

Looking at the examples of real-world problems given above, which can be interpreted as weighted college admissions problems, we find that some work analyzing the matching mechanisms in these contexts has already been done. However, only little specific attention has been paid to the special characteristics of these being *weighted* college admissions problems. For example, Veski et al. (2017) analyze the kindergarten allocation processes in two cities in Estonia and suggest different mechanisms to redesign it. One of the special features they mention but do not explicitly include in their analysis is that children with special needs in Estonia take the place of three children because they need more attention. Looking at the example of Germany, there is a law that guarantees a place in a daycare center for every child between one and three years old. However, the spots at the daycare centers are inefficiently allocated. Carlsson & Thomsen (2015) describe the current - mostly decentralized - matching mechanism in German daycare centers and show how efficiency could be improved by using the *deferred acceptance algorithm* (DA) proposed by Gale & Shapley (1962) instead of the current decentralized system. Nevertheless, they do not take into account the different staff-to-child ratios depending on the age of the children.

Another example for the weighted college admissions problem is the matching of students to supervisors for final theses. Hoyer & Stroh-Maraun (2020) analyze such a matching situation at the faculty of Business Administration and Economics at Paderborn University in Germany, where this allocation is done via a mechanism in which all students need to enroll online and state their top three preferences for potential supervisors.[2] Each supervisor has a quota of how many students he or she needs to accept. However, while each Bachelor student occupies exactly one spot at a given supervisor, each Master student occupies 1.5 spots, as their supervision is more intensive and lasts a longer time frame. Here a special situation within the weighted college admissions problem takes place, as supervisors prefer Master students, and thus students with one weight are strictly preferred over students with another weight. Hoyer & Stroh-Maraun (2020) therefore consider the two groups separately and do not analyze the impact of the weights on the mechanism itself.

Next to the empirical analyses of weighted college admission problems in the real world, also more theoretical work has been done. In the context of the standard college admissions problem with homogeneous students, Gale & Shapley (1962) show that there always exists at least one stable outcome and that one obtains a stable outcome in such a situation by applying the deferred acceptance algorithm (DA). Building on their work, there is already a large literature discussing different aspects of the college admissions problem (see, e.g., Roth 1985; Biró et al. 2010; Roth & Sotomayor 1990; Kojima et al. 2013). The weighted college admissions problem, in the literature also sometimes referred to as matching with sizes (Biró & McDermid, 2014), has also already received some attention, partly due to its relation to the hospital-residents problem with couples (HRC). In the HRC a couple of physicians submit a joint preference over hospitals in a matching problem (Roth, 1984). Interpreting a couple who has the same preference in the HRC as an agent with a weight of two in the weighted college admissions problem shows the relation between the two problems. Roth (1984) showed that a stable matching might not exist in the HRC. Roth & Rothblum (1999) describe the algorithm that is currently used in a real-world application, the NRMP, to match couples. Unfortunately, this algorithm does not necessarily find a stable outcome even if it exists (Klaus et al., 2007). There are several approaches to solve the couples problem; e.g. Biró et al. (2014) use integer programming, while Kojima (2015) adapts a

[2] A closer description of the mechanism is given in Hoyer & Stroh-Maraun (2020).

fixed-point algorithm by Echenique & Yenmez (2007). McDermid & Manlove (2010) build on this research and show that a hospital-residents problem with differently weighted residents might not result in a stable matching and that the determination whether there is one is already NP-complete. Our research, although starting with a similar question, goes in another direction. We are interested in the existence of stability in a general weighted college admissions problem and, although NP-complete, want to find an algorithm to find a stable matching if existing. Furthermore, we want to show how stability can be ensured or at least restored in weighted college admissions problems.

Dean et al. (2006) analyze a model similar to ours in the context of an ordinal transportation problem, where the authors assign different sized jobs to machines. In our terminology the different sized jobs are equivalent to students with different weights that are matched to colleges. They show that their newly proposed algorithm finds the job-optimal stable assignment among all minimally congested stable assignments. They guarantee a stable matching when the capacities of the machines are increased. We treat the increase of capacities only as a special case and develop an algorithm that finds stable matchings when they exist, even without an increase in capacities. Furthermore, we also show that decreasing the capacities is also sufficient in order to guarantee stability. There is a similar approach for the couples problem. Nguyen & Vohra (2018) show that increasing or decreasing each hospital's quota by at most two ensures finding a market where a stable matching exists. In addition, the sum of all capacities increases or decreases by at most four. As our model differs substantially from the HRC these results are not applicable to our problem.

A different approach that is more related to our own analysis here is presented by Delacrétaz (2019) who also studies stability in matching markets with sizes where the agents can have a size or weight of one or two. He proposes an algorithm that finds a stable matching if possible in this special case and relaxes the stability notion to obtain desirable results if no stable outcomes exist. Our model generalizes to arbitrary weights. Instead of relaxing the stability notion we concentrate on finding stable outcomes in this generalized model. Furthermore, we also focus on the relation to the original college admission model and the deferred acceptance algorithm.

Abizada (2016) looks at a college admissions problem where colleges offer stipends to the students while having a fixed budget. The author shows that a stable matching can still be found but not necessarily a student-optimal one. The model is related to the weighted college admissions problem as it incorporates another constraint next to the preferences. Nevertheless, as the stipends differ from college to college and from student to student the results are quite different compared to the introduction of fixed weights. Multi-dimensional constraints are introduced for the resettlement of refugees (Delacrétaz et al., 2016) as refugee families require some units of different necessities such as school places, rooms in an apartment or jobs. The authors propose a non-strategy-proof mechanism to find a stable matching, not necessarily the family-optimal one, if existing. Our model can be interpreted as handling a special case of the resettlement model, where the family has only one constraint. Already in our more simpler setting, however, we already observe the absence of strategy-proofness and optimality when identifying stable outcomes. Additionally, in contrast to the refugee resettlement model, we consider a two-sided market where both sides have preferences over the other.

Next to differently weighted participants, there are a number of other variations to the standard college admissions problem that have received some attention in the literature. One of these variations is the assumption of indifference over preferences. Erdil & Ergin (2008) show that this might not prevent the deferred acceptance algorithm from finding a stable matching, the resulting matching is, however, no longer the student- or college-optimal stable matching. Another small variation to the formulation of the college admissions problem is that students are only allowed to rank a certain number of colleges, as is the case in many real-life applications (Pathak & Sönmez, 2013). Haeringer & Klijn (2009) show that these constrained rank ordered lists cause manipulability of the DA and thus stability might not be ensured any more. Overall, the DA is thus not robust to small variations in the formulation of the matching problem. Our findings are in line with these results.

The paper is structured as follows. In Section 2, we describe the weighted college admission model. In Section 3, we then analyze stability in a setting where students are weighted differently. We first look at the case where colleges have strict preferences for one type of student. Following from this we analyze the case where there are no strict preferences, showing that there are not always stable outcomes in this market and that even if there are stable outcomes the DA does not necessarily find them. Finally, we introduce the DAG and analyze its properties. Afterwards, we discuss how stability can be ensured (Section 4) and conclude in Section 5.

2 Model

To model the weighted college admissions problem, we use as a baseline model the general college admissions problem by Roth & Sotomayor (1990) and also follow their terms and notation. Therefore, to keep the language coherent, we will from now on refer to the matching situation as a college admissions problem and to the two market sides as colleges and students throughout the paper. To include the weights, we assume that students can have different weights at colleges and thus take up different amounts of capacity.

Formally, we assume we have two finite and disjoint sets $C = \{c_1, ..., c_n\}$ and $S = \{s_1, ..., s_m\}$ of colleges and prospective students, respectively. We additionally assume that the preferences P on both sides are complete and transitive so that they can be represented by ordered lists. We denote by $c \succ_s c'$ that student s prefers college c over college c' and by $s \succ_s c$ that student s prefers to remain unmatched over being matched with college c (in other words, college c is not acceptable for student s). Similarly, $s \succ_c s'$ denotes that college c prefers student s over student s' and $c \succ_c s$ denotes that s is unacceptable. We indicate the preferences of a student s as $P(s)$ and the preferences of college c as $P(c)$. For each college c there is a fixed quota or capacity q_c of how many slots it actually offers.

To implement the idea of differently weighted students in the model, we assume that some students may take up more than one slot at a given college and thus have different weights. Therefore, each student has a weight p_s, where $p_s \geq 1$. Student $\underline{s}$ has the smallest weight $p_{\underline{s}} = 1$ and student $\overline{s}$ has the highest weight $p_{\overline{s}} = a \cdot p_{\underline{s}}$, where $a \in \mathbb{R}_{>1}$. If more than one student has the same weight, we say that all students with the same weight are of the same type. There are d different weights or types, $T = \{1, \ldots, d\}$ with $d \leq m$. The family of sets $T^1, \ldots, T^T$ is a partition of the set of students S in such a way that all students s that are elements of T^t are of type t and have the same weight, denoted as p_{s^t}. Thus, a student s^t is of type t and has the weight p_{s^t}. If there are just two types of students, the problem reduces to students $\underline{s}$ who take up exactly one slot at a given college ($p_{\underline{s}} = 1$) and students $\overline{s}$ who take up $p_{\overline{s}} = a \cdot p_{\underline{s}}$ slots at any given college. Note that in the classical model, all students are assumed to have the same weight $p_s = 1 \;\; \forall \, s \in S$. An outcome of the (weighted) college admission process assigns each student to at most one college and each college is matched to a set of students that jointly do not exceed the college's capacity. A matching is a mapping μ on $C \cup S$ with $\mu(c) \subseteq S$ for all $c \in C$ and $|\mu(c)| \leq q_c$, $\mu(s) \in C \cup \{s\}$ for all $s \in S$, and $\mu(s) = c$ if and only if $s \in \mu(c)$. If a student stays unmatched we denote this as $\mu(s) = s$.

We follow the concept of stability in college admissions problems as was formulated by Roth & Sotomayor (1990). They show that a matching is stable if it is individually rational and there are no blocking pairs. However, as colleges will often have capacities that allow for more than one student, colleges might also compare different matchings with each other, meaning preferences over different groups or sets of students. For tractability, in the original college admissions problem it is therefore assumed that colleges' preferences are responsive, which means that the preference over outcomes corresponds to the preference over individual students. Thus, it is sufficient to focus on pairwise stability instead of group stability (Roth, 1985). We adjust this assumption here to also account for weights, by assuming that colleges' preferences are responsive with weights. The crucial difference is that we introduce a feasibility condition that the two matchings that are compared are both not exceeding the college's capacity. As students have different weights, it might happen that by replacing a student in a matching with another student, the quota is exceeded as the new student has a greater weight than the former one. A matching that exceeds the capacity is not a feasible outcome. The concept is stronger and more restrictive and thus reduces to the original notation of responsiveness if all students are weighted equally. Thus, we follow the original model by considering a special version of pairwise stability.

Definition 1 (Responsiveness with Weights)**.** *The preference relation* $\tilde{P}(c)$ *of college* c *over sets of students is* responsive with weights *to the preferences* $P(c)$ *of college* c *over individual students if, whenever* $\mu'(c) = \mu(c) \cup \{s_k\} \backslash \{\sigma\}$ *for* $\sigma \in \mu(c)$*,* $s_k \notin \mu(c)$*,* $|\mu(c)| \leq q_c$*, and* $|\mu'(c)| \leq q_c$*, then* c *prefers* $\mu'(c)$ *to* $\mu(c)$ *under* $\tilde{P}(c)$ *if and only if* c *prefers* s_k *to* σ *under* $P(c)$*.*

Using responsive preferences, Roth & Sotomayor (1990) define a matching μ as stable "if it is not blocked by any individual agent or any college-student pair." To account for different weights in this concept, we need to adjust the definition of a blocking pair, whereas individual rationality remains the same.

Definition 2 (Individual Rationality (Roth & Sotomayor, 1990))**.** *A match* $\mu(s) = c$ *implies that* s *and* c *find each other acceptable.*

Individual rationality in this context means that both, college and student, prefer to be matched to one another over remaining unmatched. To define a stable matching accounting for weighted students, we first define a weighted blocking pair. Here, we call q_c^μ the remaining quota of a college c in a matching μ.

Definition 3 (Weighted Blocking Pair)**.** *A matching* μ *is blocked by a pair* (c, s) *if* $\mu(s) \neq c$ *and* $c \succ_s \mu(s)$*. Additionally, it has to hold that either* $s \succ_c s'_l$ *for all* $l \in \{1, \ldots, k\}$ *where* $\{s'_1, \ldots, s'_k\} \subseteq \mu(c)$ *with either* $p_s \leq q_c^\mu - \sum_{l=1}^{k} p_{s'_l}$ *or* $s \succ_c c$ *with* $p_s \leq q_c^\mu$*.*

Thus, a pair blocks the allocation if the student prefers the college over his or her current match and the college prefers the student over every single student in a set of students that is currently matched to this college such that it would have enough remaining quota to be matched to the student without this set. It is also sufficient if the student prefers the college over his or her current match and the college's remaining quota is large enough for a student to be placed in the college. Please note that this is only an extension of the definition without weights. If all students $s \in S$ have the same weight p_s, the condition reduces to the original definition of blocking pairs given in Roth & Sotomayor (1990).

Definition 4 (Weighted Stability)**.** *A matching is weighted stable if it is individually rational and there exists no weighted blocking pair.*

Again, the definition of weighted stability reduces to the original definition of stability if the weight p_s is the same for all students $s \in S$.[3] If different weights play a role, this definition ensures that the colleges still try to fill their quotas while taking their preferences into account. This definition is equivalent to the one of McDermid & Manlove (2010). For readability, we will use the term, stability, from now on instead of, weighted stability.

3 Stability in Weighted College Admissions Problems

To analyze the stability of a matching we look at two different cases. To start with, we assume that all colleges strictly prefer students with a specific weight over students with another weight. Later, we will drop this assumption.

3.1 Strict Preference for One Type of Students

We begin our analysis by assuming that all colleges strictly prefer students with a specific weight over all students with another specific weight. Thus, colleges have a preference order over different weights, and within each group they have preferences over the individual students. To fix ideas, assume we have only two types of students and the colleges strictly prefer students $\overline{s}$ with weight $p_{\overline{s}}$ over students $\underline{s}$ with weight $p_{\underline{s}}$. Thus, even the least preferred student of type $\overline{s}$ is still ranked higher by each college than the most preferred student

[3] As both the definition of blocking pairs as well as the definition of stability reduce to their original form as given in Roth & Sotomayor (1990), when all students have the same weight, we do not state the original conditions here explicitly but refer the interested reader to Roth & Sotomayor (1990).

of type $\underline{s}$.[4] To capture the colleges' preferences over student types we assume that colleges have a common type preference. Therefore, we assume that each college c has original preferences which are independent of the students' types $P(c)$ and a preference ordering of the types t. We assume w.l.o.g. that students of type 1 are the most preferred type, type 2 are the second most preferred student type, and students of type j are the j most preferred type for $j \geq 1$.

Definition 5. *We construct college's c's type preference $P^t(c)$ from its original priorities $P(c)$ as follows: For all $s, s' \in S$ with $s \in T^t$, $s' \in T^{t'}$*

$$sP^t(c)s' \quad \textit{if and only if} \quad \textit{(a) } t < t' \textit{ or (b) } t = t' \textit{ and } sP(c)s'. \tag{0.1}$$

As the ordering of types is the same for all colleges, the colleges have a common type preference.

In such a case, we can simply regard the different types of students as separate matching markets where first only the most preferred type is allocated, then the next preferred type is allocated to the remaining free slots. This continues until either no college has free capacity left or all students are matched.

Proposition 1. *Given is a college admissions problem with students $s_1, \ldots, s_m$ of types t and corresponding weights p and colleges $c_1, \ldots, c_n$ with quotas q. If all colleges have a common type priority, a stable matching can be obtained.*

Proof. Without loss of generality we assume that students of type s^1 with weights p_{s^1} are the most preferred type, the type s^2 with weights p_{s^2} are the second most preferred type of students and generally students of type s^j are the j most preferred type for $j \geq 1$. We can apply the deferred acceptance algorithm (DA) per type, starting with the students of T^1. From Gale & Shapley (1962), we know that the DA yields a stable allocation. After the first DA with students s^1, all student college matches are removed and the capacities of the colleges are reduced accordingly. If a college c has some free

[4] Of course the argument also works when colleges prefer students with a low weight over those with higher weights.

capacity left after this first DA, the quota of college c is either big enough that also at least one student of the remaining $(d-1)$ types with the smallest weight $p_{\underline{s}}$ might still be accepted, thus $q_c^\nu \geq p_{\underline{s}}$, or it is not, thus $q_c^\nu < p_{\underline{s}}$. If the quota is not large enough the college completely exits the mechanism, otherwise it remains with its reduced quota. A next round of the DA can then be run on students of type s^2. This continues until we have applied the DA over the least preferred group of students. A lower type student s^{j+l} with $l \geq 1$ is never able to block a pair of a student s^j and a college c as she is always less preferred by c, $s^j \succ_c s^{j+l}$ for all j and l. Therefore, as all allocations for each type are stable individually, the combined matching is stable for all students. □

Consequently, if students are strictly preferred according to their weights, running the DA multiple times leads to a stable outcome and all the properties of the DA are kept intact.

3.2 Weighted College Admissions Problems without Stable Matchings

If there is no clear preference ranking of students according to their weights, each college faces making a more complicated decision. To decide whether a student is rejected, a college has to consider not only its preferences but also its quota and the student's weight. In the following section we will analyze whether in such a setting there exists a stable matching and how to reach it in case of existence.

In the classical model where students all have the same weight, there always exists a stable outcome (Gale & Shapley, 1962). Unfortunately, when students are weighted differently there are cases in which there is no stable matching at all. Looking at Example 1, we see that here we always find a blocking pair with weights and thus no stable matching exists.

Example 1. *Assume there are three colleges c_1, c_2 and c_3 with quotas $q_1 = 1.5$, $q_2 = 2$, and $q_3 = 1.5$ and preferences over students $P(c)$. There are three students: two Bachelor students b_1 and b_2 with preferences $P(b)$ and one*

Master student m_1 with preferences $P(m_1)$. The Bachelor students b_1 and b_2 are of type $\underline{s}$ and have weight $p_{\underline{s}} = 1$ and the Master student m_1 is of type $\overline{s}$ and has weight $p_{\overline{s}} = 1.5$. The preferences of the three students and the colleges are as follows:

$$
\begin{array}{lll}
P(b_1) : c_1,\, c_2,\, c_3 & P(m_1) : c_2,\, c_3,\, c_1 & P(c_1) : b_2,\, b_1,\, m_1 \\
P(b_2) : c_2,\, c_1,\, c_3 & & P(c_2) : b_1,\, m_1,\, b_2 \\
 & & P(c_3) : m_1,\, b_2,\, b_1
\end{array}
$$

To see that there is no stable matching in this example, let us consider the potential allocations. In total there are eight different feasible allocations in which all students are matched. These are all given in the left column of Table 1. Given their quotas, colleges c_1 and c_3 might accept either one Bachelor student or one Master student. College c_2 can accept at most either two Bachelor students or one Master student. It is easy to see that there always exists a blocking pair when considering the different weights of students. In the right column of Table 1, we show one of potentially multiple blocking pairs that blocks the allocation in the left column. As there are no unacceptable matches, all other feasible allocations in which not all students are matched are not stable either as the unmatched students would always find colleges with remaining capacities that are willing to take them in. Thus, it is straightforward to see that for every potential allocation in this problem there exists at least one blocking pair. Therefore, by definition, there does not exist a stable allocation.

As in Example 1 we do not find a stable allocation in a weighted college admissions problem with two differently weighted groups of students, we have thus shown that there are cases in which no stable outcome exists. Proposition 2 directly follows from this.

Proposition 2. *When there are two or more types of students with different weights p_s, there are cases in which no stable outcome exists.*

This result poses a marked distinction to matching markets without differently weighted subjects, where stable matchings always exist. Here, on the other hand, there are clearly cases in which no stable matching exists. The

Feasible Allocations	Blocking Pair
(c_1, b_1), (c_2, b_2), (c_3, m_1)	(c_2, m_1)
(c_1, b_2), (c_2, b_1), (c_3, m_1)	(c_2, b_2)
(c_1, b_1), (c_2, m_1), (c_3, b_2)	(c_1, b_2)
(c_1, b_2), (c_2, m_1), (c_3, b_1)	(c_2, b_1)
(c_1, m_1), (c_2, b_1), (c_3, b_2)	(c_1, b_1)
(c_1, m_1), (c_2, b_2), (c_3, b_1)	(c_1, b_1)
(c_1, m_1), (c_2, b_1), (c_2, b_2)	(c_1, b_1)
(c_2, b_1), (c_2, b_2), (c_3, m_1)	(c_1, b_1)

Table 1: Feasible allocations and blocking pairs in Example 1

result that no stable matching may exist in markets with differently weighted participants has also been shown by McDermid & Manlove (2010). They analyze a special case of the hospital residents matching with couples, in which members of a couple have individual preference lists over hospitals and these lists are consistent with the joint preference list of the couple. They show that in this case a stable matching does not always exist. The special case that a couple wants to attend the same hospital, and therefore occupies more than one position there, can also be interpreted as residents with different weights applying to a hospital. The result that in this case a stable matching does not always exist is therefore analogous to our result here. Delacrétaz (2019) also uses this analogy in his paper on matching with sizes to show that stability is no longer guaranteed.

3.3 Weighted College Admissions Problems with Stable Matchings

In the previous section, we have shown that if weights are introduced to the standard college admissions problem, stable allocations do not always exist. In the next step we are looking at a weighted college admissions problem in which a stable matching exists and we analyze the performance of the DA. We show that opposed to the standard college admissions problem, here the DA does not perform satisfactorily, as it does not result in a stable outcome.

Example 2. *Assume there are three colleges c_1, c_2 and c_3 with quotas $q_1 = 3$, $q_2 = 5$ and $q_3 = 2$ and preferences over students $P(c)$. There are seven students: four Bachelor students b with preferences $P(b)$ and three Master students m with preferences $P(m)$. The Bachelor students b_1, b_2, b_3 and b_4 are of type $\underline{s}$ and have weight $p_{\underline{s}} = 1$ and the Master students m_1, m_2 and m_3 are of type $\overline{s}$ with weight $p_{\overline{s}} = 2$. The preferences of the students and the colleges are as follows:*

$P(b_1) : c_1, c_2, c_3$ $\quad P(m_1) : c_2, c_1, c_3$ $\quad P(c_1) : b_3, b_2, m_3, b_1, b_4, m_1, m_2$
$P(b_2) : c_1, c_3, c_2$ $\quad P(m_2) : c_2, c_1, c_3$ $\quad P(c_2) : b_4, b_2, m_1, b_1, m_2, b_3, m_3$
$P(b_3) : c_2, c_1, c_3$ $\quad P(m_3) : c_1, c_3, c_2$ $\quad P(c_3) : b_1, m_3, b_2, b_4, m_2, m_1, b_3$
$P(b_4) : c_2, c_3, c_1$

	c_1	c_2	c_3
Potential matching η	b_2, b_3	b_1, b_4, m_1	m_3
Remaining quota q_c^η	1	1	0

Table 2: Assignment of students to colleges using the DA

Using the DA to find a matching μ in this example results in the outcome shown in Table 2. The DA here works as follows. If the remaining quota is smaller than the weight of the newly proposing student, the college compares the student to the least preferred student who has already proposed and is not rejected yet. Depending on the type of the newly proposing student, this comparison takes one of two forms. If the newly proposing student is of type $\underline{s}$, the college simply compares the student to the least preferred student who has already proposed and is not rejected yet. If they prefer the newly proposing student, he is tentatively assigned a place at the college. If, however, the newly proposing student is of type $\overline{s}$, the college again compares him to the least preferred student who has already proposed and is not rejected yet. If this student also has a weight of $p_{\overline{s}}$ or is preferred to the newly proposing student, the college just decides according to its preferences which of the two students to reject. If the least preferred student s has a weight of $p_{\underline{s}} < p_{\overline{s}}$ and is not preferred to the newly proposing student, it may still be the case that there *is not enough capacity left at the college to accept the proposing student of type* $\overline{s}$*. Then the college needs to compare the proposing student to the next least preferred student. This continues until either the student is tentatively assigned a place at the college or he is rejected. He is tentatively assigned if there are enough students that are less preferred than him so that there is enough capacity for him. All the other students are then rejected. If there are not enough students who are less preferred, the newly proposing student is rejected.*

Obviously, the resulting matching shown in Table 2 is not stable. Student b_1 *would rather attend college* c_1*, and* c_1 *has enough remaining capacity left. Thus,* b_1 *and* c_1 *form a blocking pair and the matching is not stable. We reach this matching using the DA because initially* b_1 *is rejected at* c_1 *in favor of* m_3*, who is later on rejected in favor of* b_3*. This leaves an open capacity of* 1 *at* c_1*, which is not filled and student* b_1 *has no possibility to propose again to college* c_1*.*

Although not finding it with the DA, stable matchings exist in the matching market in Example 2; e.g., the one in Table 3.

	c_1	c_2	c_3
Final matching μ	b_1, b_2, b_3	b_4, m_1, m_2	m_3
Remaining quota q_c^μ	0	0	0

Table 3: Stable matching for Example 2

Consequently, we can conclude the following. When there are two or more types of students with different weights, the deferred acceptance algorithm does not always lead to a stable matching μ, even if μ exists.

In general, in cases with differently weighted students, the DA may not find an existing stable outcome. This is due to the fact that in contrast to the original problem students who are rejected by a college could sometimes be accepted again at a later point in time, when a student with a higher weight is rejected and room for a student with a lower weight appears. Imagine there are two types of students $\overline{s}$ and $\underline{s}$ with $p_{\overline{s}} \geq p_{\underline{s}}$. A student $\underline{s}_i$ is rejected at a college c in favor of student $\overline{s}_t$, who is then rejected later in favor of student

$\underline{s}_j$ leaving the college with free capacity but no possibility for $\underline{s}_i$ to propose again. Instead, $\underline{s}_i$ and the college form a blocking pair (see Example 2). To overcome this and similar issues, we propose to add a system of gaps to the DA mechanism, leaving everything else unchanged.

3.4 The Student-proposing DA with Gaps

To design a new mechanism that will function well in a setting with differently weighted students, we start by using the standard DA implementing the procedure for the different weights of students as described above. However, to overcome the difficulties outlined above, we introduce the concept of gaps. To give an intuition why our proposed mechanism works, whereas the standard DA does not necessarily reach a stable outcome, think of the following situation. In the student-proposing DA there is a college c and a preliminary matching μ in which some student s is matched to c and c has a remaining quota of $q_c^\mu \geq 0$. Now a new student s' proposes to college c. If s' has a lower weight than student s and is tentatively accepted instead of s, the remaining quota q_c^ν increases in the new matching ν compared to q_c^μ. Such an increase in remaining capacity cannot occur in the standard college admissions problem and it is the reason why the DA does not work satisfactorily in a problem with different weights. To overcome this, we introduce the deferred acceptance algorithm with gaps (DAG), which allows students to propose again to colleges that rejected them previously if such an increase in remaining capacity occurs. With this feature we are actually able to find stable matchings if they exist.

Definition 6 (Gap)**.** *In the DAG a* gap g_c^μ *denotes an increase in the remaining quota* q_c^μ *of college* c *with the temporary matching* μ *compared to some previous temporary matching* ν *and its remaining quota* q_c^ν*; more precisely,* $g_c^\mu = q_c^\mu - q_c^\nu$*. A gap also always occurs if a student leaves college* c *voluntarily to propose to a gap at college* c'*, where she was previously rejected and is now tentatively accepted. The vector* g^μ *collects all colleges' gaps in round* μ*.*

There are thus two possibilities for gaps to occur. First a gap occurs if there is an increase in the remaining quota. This occurs if one or more students who were tentatively accepted at college c in the previous temporary matching ν are rejected in matching μ and there is a difference between their weight and the weight of the newly tentatively accepted students in matching μ. Second, a gap also occurs if a student leaves a college voluntarily. In this case, we might not observe an increase in the remaining quota as another student might replace the voluntarily leaving student immediately. Nevertheless, we say a gap occurs here. Following from our definition of a gap, we now introduce a new algorithm based on the DA that incorporates gaps to find stable matchings in a weighted college admissions problem, the so-called student-proposing *DA with Gaps (DAG)*. It works as follows.

Step 1. Each student proposes to her first choice. Each college rejects any unacceptable proposals and tentatively accepts students according to its preferences and capacity.

Step 2. Each student who was rejected in Step 1 proposes to her most-preferred college that has not yet rejected her. As in Step 1 each college rejects unacceptable offers, tentatively accepts proposals according to its preferences and capacity, and rejects the rest. There may occur *gaps*. All colleges with gaps are marked. This mark stays until the gap is triggered.

Step 3. If any gaps have occurred previously, one randomly chosen marked college, $\hat{c}$, is triggered. Each student that has already been rejected by $\hat{c}$, except for the one who caused the gap,[5] proposes again to $\hat{c}$. Afterwards, the mark is deleted.[6] All other students who were rejected in the previous step propose to the next most-preferred college on their preference lists. Colleges consider all proposals at the same time (independent of whether they are due to a triggered gap or not). They reject unacceptable offers, tentatively accept proposals according to their preferences and capacities, and reject the rest. If

[5]If the gap is not triggered directly, it may be increased by other students leaving this college. Then all the students who caused parts of the gap are allowed to propose again once the gap is triggered.

[6]Please note that in each step only at most one mark is triggered. All other marked colleges stay marked until they are triggered in one of the following steps.

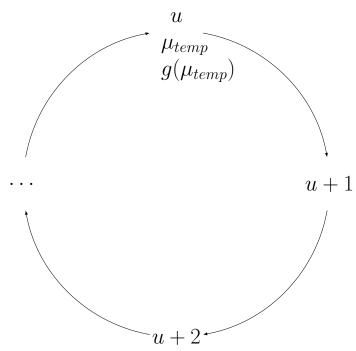

Figure 1: Illustration of a Cycle

a student who proposes to a gap is rejected immediately, she returns to her former match if she is not rejected immediately there as well. New gaps may occur. Step 3 is repeated until the algorithm terminates.

The algorithm terminates when either there are no more gaps to be triggered *and* no more rejections or when a cycle (as defined in Definition 7 and illustrated in Figure 1) occurs.

Definition 7. *A* cycle *in the DAG occurs if and only if in round u the exact same temporary allocation μ_{temp} and the exact same gaps occur as in some round v with $u < v$. Additionally, independent of the order of the triggered gaps in the rounds between u and v, we always come to a round v, where everything is exactly the same as in round u.*

With the help of the following Example 3 we illustrate the functioning of the DAG if there is a stable outcome.

Example 3. *Assume there are two colleges, c_1 and c_2, with quotas $q_1 = 3$ and $q_2 = 2$ and preferences over students $P(c)$. There are five students: three Bachelor students b with preferences $P(b)$ and two Master students m with preferences $P(m)$. The Bachelor students b_1, b_2 and b_3 are of type $\underline{s}$ and have weight $p_{\underline{s}} = 1$ and the Master students m_1 and m_2 are of type $\overline{s}$ with weight $p_{\overline{s}} = 2$. The preferences of the students and the colleges are as follows:*

$$P(b_1) : c_1,\, c_2 \qquad P(m_1) : c_1,\, c_2 \qquad P(c_1) : b_3,\, b_2,\, m_1,\, b_1,\, m_2$$

$P(b_2) : c_1, c_2 \qquad P(m_2) : c_2, c_1 \qquad P(c_2) : m_2, b_1, b_3, m_1, b_2$
$P(b_3) : c_2, c_1$

To illustrate how the DAG works, we go through Example 3 round by round (Table 4). In each round, students are tentatively accepted (μ^{temp}). All gaps are marked with a square. Only one marked college is triggered at a time. It can easily be checked that the resulting matching is indeed stable.

The algorithm starts like the DA with the students proposing to their most-preferred colleges and the colleges rejecting students according to their preferences and capacities. In Round 2 a gap at c_1 occurs as m_1, who was tentatively accepted before, is rejected in favor of b_3, leaving c_1 with a greater remaining quota than before. Thus, c_1 is marked and in Round 3 this mark is triggered. Student b_1, who has been rejected before, is allowed to propose again to c_1. As c_1 has enough remaining quota for b_1, it accepts her. All other students keep proposing according to the DA. This means that m_1 now proposes to c_2 and is rejected there. The algorithm terminates after Round 3 as m_1 was rejected by all colleges and thus cannot propose anymore, while the other students are tentatively accepted. The students b_1, b_2 and b_3 are matched to c_1, m_2 is matched to c_2. m_1 stays unmatched.

Round		College	
		c_1	c_2
1	proposing	b_1, b_2, m_1	b_3, m_2
	μ^{temp}	b_2, m_1	m_2
2	proposing	b_2, b_3, m_1	b_1, m_2
	μ^{temp}	$b_2, b_3, \square$	m_2
3	proposing	b_1, b_2, b_3	m_1, m_2
	μ^{temp}	b_1, b_2, b_3	m_2
Final matching μ		b_1, b_2, b_3	m_2
Remaining quota q_c^μ		0	0

Table 4: Assignment of students to colleges in Example 3 using the DA with Gaps

3.5 Properties of the DAG

We have seen in the previous example that the algorithm terminates in a stable matching. Additionally, we can show that if and only if there is a stable matching, the algorithm terminates once all students are allocated to a college or are rejected by all their acceptable colleges. Otherwise, the procedure cycles. Let us start by looking at the case when a matching is found by the DAG.

Theorem 1. *Any resulting matching μ of the deferred acceptance algorithm with gaps (DAG) is stable.*

Proof. If the student-proposing DAG terminates, it results in a matching μ. If a student s prefers college c over her current match $\mu(s)$, thus $c \succ_s \mu(s)$, this means that s was already rejected by c during the mechanism as all students propose according to their preferences. The rejection implies that, as long as s is acceptable for c, c preferred some other student s' over s, thus $s' \succ_c s$, and is either still matched to this other student or to an even more preferred student. By favoring s' the remaining quota has been too small for s to fit in. Of course, it is possible that c rejected s multiple times as a subsequent rejection of other students might have opened a slot for s again. Nevertheless, as s and c are not matched in μ, c always rejected s in favor of another student even if new slots had opened.

Thus, each student s' matched to c in $\mu(c)$, thus $s' \in \mu(c)$, is either preferred over s, thus $s' \succ_c s$, or has a smaller weight than s with $p_{s'} < p_s$. Additionally, in the latter case it holds for all $s' \in \mu(c)$ with $s \succ_c s'$ that $p_{s'} < p_s$, $\sum p_{s'} < p_s$ and $q_c^\mu - \sum p_{s'} < p_s$. Therefore, c and s cannot form a blocking pair. Either c prefers the students currently matched to it over s or, if c prefers s over some students that are currently matched to it, c has not enough free capacity for s to fit in even if all these students would not be matched to it.

If c prefers another student s'' who is not matched to c over one or more currently matched students with sufficient weight, s'' prefers her current match over c. Thus, they cannot form a blocking pair either.

Therefore, any resulting matching μ of the DAG is stable. □

The mechanism thus produces stable outcomes if it results in a matching.

Additionally, if the DAG cycles this means that there is no stable outcome.

A cycle may only occur if at least one gap exists and by triggering this gap at least one student, who proposes again, is tentatively accepted by the college, causing another gap to occur in the mechanism (at some point). Thus, in the algorithm and also particularly in the cycle, a student might leave a college either if she is rejected or if she is trying to fill a gap at another college. In both cases the temporary matching was not stable as either the college that rejected the student or the student herself would have blocked it.

Theorem 2. *If the mechanism cycles, there is no stable outcome.*

Proof. If at no time a gap occurs, the DAG works exactly as the original DA, which never cycles and will always conclude with a stable outcome (Gale & Shapley, 1962). The same holds if only one gap occurs in the mechanism. In that case all students who were tentatively accepted at another college and who can propose to this gap will directly be rejected by the college and return to their previous allocation. Therefore, no other gaps occur and the mechanism continues like the regular DA. If a student who was currently not temporarily allocated to any other college proposes to the gap, he is either accepted or rejected, but no other gap is induced by this. Again, the mechanism continues like the regular DA. A cycle is thus always induced by the existence of at least two gaps. If and only if at least two gaps occur in the course of the mechanism, the DAG may cycle and therefore not terminate.

Imagine the algorithm cycles but there is a matching ν that is stable but never reached. First, ν cannot be found inside a cycle, as one student is always rejected for a gap to occur. Inside the cycle, students keep moving from one college to another either when they are rejected or when there is a gap that they want to fill. More precisely, each student is proposing at every step of the algorithm according to her preference list and is eventually rejected until she has reached the end of her list. In this case she will start proposing again only if there is a gap.

Second, there is also no stable outcome outside the cycle. Imagine we are in a cycle and there is some matching ν. As there is at least one student who is rejected or voluntarily leaving (and thus causing a new gap), there is always a student s proposing according to her preferences and so has to reach the stable match $\nu(s)$ at some point. But in this case, there would not be a cycle. Thus, if there is a cycle, there is no stable outcome inside or outside the cycle. □

With help of the continued Example 1, we want to illustrate what happens in the DAG if there is no stable matching. Instead of resulting in a matching, the algorithm cycles.

Example 1 (continued). *From the previous analysis of Example 1 we already know that this example has no stable matching. In Table 5, we illustrate how the DAG works in this case. We do not find a final matching here, instead the mechanism cycles. The beginning and end of the cycle is marked in gray. There is no chance to change the order in which the marked colleges are triggered to leave the cycle either. Please note that in Rounds 4, 5 and 6 students appear twice in the table because they propose to colleges due to gaps that are triggered and they apply to these colleges. However, they have not been rejected at the other college they were tentatively assigned to in the previous round. More precisely, in Round 4 student b_2 proposes again to college c_2 as a gap is triggered there. As student b_2 was tentatively accepted before at c_1, she also has the possibility to stay there if c_2 would reject her. Therefore, we denote this possibility by b_2^*. As the student is tentatively accepted in Round 4 at college c_2 she gives up her seat at c_1, voluntarily causing a gap there. Similarly, in Round 5 student b_1 proposes to c_1 while being temporarily matched to c_2 and in Round 6 student m_1 proposes to a gap in c_2 while being tentatively accepted at c_3.*

So far we have analyzed what happens if there is one stable matching (Example 3) or if there is no stable matching (Example 1). Now imagine that there are two or more stable outcomes. In this case the order in which marked colleges are triggered may influence the outcome of the DAG. To illustrate this let us look again at Example 2. The outcome of the DAG is always stable (see Theorem 1), but we may find more than one stable matching by alternating the order in which marked colleges are triggered.

Example 2 (continued). *Again we list the separate rounds and gaps for*

Round		College		
		c_1	c_2	c_3
1	proposing	b_1	b_2, m_1	
	μ^{temp}	b_1	m_1	
2	proposing	b_1, b_2	m_1	
	μ^{temp}	b_2	m_1	
3	proposing	b_2	b_1, m_1	
	μ^{temp}	b_2	b_1, $\square$	
4	proposing	b_2^*	b_1, b_2	m_1
	μ^{temp}	$\square$	b_1, b_2	m_1
5	proposing	b_1	b_1^*, b_2	m_1
	μ^{temp}	b_1	b_2, $\square$	m_1
6	proposing	b_1	b_2, m_1	m_1^*
	μ^{temp}	b_1	m_1	$\square$
7	proposing	b_1, b_2	m_1	
	μ^{temp}	b_2	m_1	

Table 5: The student-proposing DAG cycles when there is no stable matching (Example 1)

the DAG in Table 6 and Table 7. After the first two rounds, colleges c_1 and c_2 are both marked with a gap. Randomly, one of the two marked colleges is now triggered. In Table 6 the gap at c_1 is triggered first. In this case, c_2 stays marked until the gap is triggered in Round 4. As in the meantime Bachelor student b_1 proposes again to c_1 and is tentatively accepted, c_2's gap gets bigger. In Round 4, c_2's now bigger gap is triggered and b_3 and m_2 propose again. As b_3 is immediately rejected, she returns to her temporary match c_1 again. To denote this, student b_3 appears twice in Round 4. She proposes to the gap that was triggered at c_2 while securing her former match c_1 denoted by b_3^*. It is straightforward to see that the resulting matching μ is indeed stable. c_1 is matched to b_1, b_2 and b_3, c_2 is matched to b_4, m_1 and m_2, and m_3 is placed in college c_3.

Alternatively, after Round 2 we can alter the order of the new proposals to find another stable matching ν by triggering the gap at c_2 first (cf. Table 7). In the resulting matching b_2 and m_3 are matched to c_1, b_1, b_3, b_4 and m_1 are

Round		College		
		c_1	c_2	c_3
1		b_1, b_2, m_3	b_3, b_4, m_1, m_2	
	μ^{temp}	b_2, m_3	b_4, m_1, m_2	
2		b_2, b_3, m_3	b_1, b_4, m_1, m_2	
	μ^{temp}	b_2, b_3, $\square$	b_1, b_4, m_1, $\square$	
3		b_1, b_2, b_3, m_2	b_1^*, b_4, m_1, $\square$	m_3
	μ^{temp}	b_1, b_2, b_3	b_4, m_1, $\square$, $\square$	m_3
4		b_1, b_2, b_3^*	b_3, b_4, m_1, m_2	m_3
	μ^{temp}	b_1, b_2, b_3	b_4, m_1, m_2	m_3
μ		b_1, b_2, b_3	b_4, m_1, m_2	m_3
q_c^μ		0	0	0

Table 6: First assignment of students to colleges using the DA with Gaps in Example 2

placed into c_2 and m_2 is offered a slot at c_3.

As we have seen in the previous example, we can sometimes find more than one stable matching. It is easy to see that the second matching ν leaves the students overall worse off than the matching μ. In both matchings one Bachelor student gets her second choice while the others get their first choices. Furthermore, two out of three Master students get their first choices in both outcomes, and while in matching μ the remaining Master student is matched to her second choice, in matching ν the remaining student gets her third choice. Consequently, while the DAG always leads to a stable outcome if it exists, not each stable matching that the DAG yields is student-optimal. This is a marked difference from the regular DA, which always yields the student-optimal stable matching.

Corollary 1. *The resulting matching μ of the student-proposing DAG is not necessarily the student-optimal stable matching.*

Furthermore, by looking at the previous example we find the following. Student b_3 can actually influence the algorithm in such a way that she can guarantee her personally more preferable result.

Round		College		
		c_1	c_2	c_3
1		b_1, b_2, m_3	b_3, b_4, m_1, m_2	
	μ^{temp}	b_2, m_3	b_4, m_1, m_2	
2		b_2, b_3, m_3	b_1, b_4, m_1, m_2	
	μ^{temp}	b_2, b_3, $\square$	b_1, b_4, m_1, $\square$	
3		b_2, b_3^*, m_2, $\square$	b_1, b_3, b_4, m_1	m_3
	μ^{temp}	b_2, m_2, $\square$, $\square$	b_1, b_3, b_4, m_1	m_3
4		b_1, b_2, m_2, m_3	b_1^*, b_3, b_4, m_1	m_3^*
	μ^{temp}	b_2, m_3	b_1, b_3, b_4, m_1	$\square$
5		b_2, m_3	b_1, b_3, b_4, m_1	m_2
	μ^{temp}	b_2, m_3	b_1, b_3, b_4, m_1	m_2
Final matching ν		b_2, m_3	b_1, b_3, b_4, m_1	m_2
Remaining quota q_c^ν		0	0	0

Table 7: Second assignment of students to colleges using the DA with Gaps in Example 2

Example 2 (continued). *Imagine that student b_3 wants to ensure her more favorable outcome ν. Therefore, the student manipulates her stated preferences by not stating c_1 anymore:*

$P‘(b_3)$: c_2, c_3.

With this manipulation b_3 can prevent the students b_1 and m_2 from proposing again to c_1, as m_3 is not rejected by c_1 in favor of b_3. Thus, no gap is marked. Instead, b_3 is allowed to propose again at college c_2 and gets her true first choice in the end (see Table 8).

Thus, as long as no gaps exist, the DAG works in the same way as the original DA. In this case, all the properties of the DA hold. Nevertheless, as soon as gaps exist, we cannot guarantee for strategy-proofness or optimality anymore even if we limit the considerations to cases where a stable matching exists. We can conclude that there are incentives for students to strategize the stating of preferences in some cases. In contrast to the original DA, unfortunately we can no longer ensure strategy-proofness.

Corollary 2. *The DAG is not strategy-proof for the proposing agents if gaps exist.*

Round		College		
		c_1	c_2	c_3
1	proposing	b_1, b_2, m_3	b_3, b_4, m_1, m_2	
	μ^{temp}	b_2, m_3	b_4, m_1, m_2	
2	proposing	b_2, m_3	b_1, b_4, m_1, m_2	b_3
	μ^{temp}	b_2, m_3	b_1, b_4, m_1, $\square$	b_3
3	proposing	b_2, m_2, m_3	b_1, b_3, b_4, m_1	b_3^*
	μ^{temp}	b_2, m_3	b_1, b_3, b_4, m_1	$\square$
4	proposing	b_2, m_3	b_1, b_3, b_4, m_1	m_2
	μ^{temp}	b_2, m_3	b_1, b_3, b_4, m_1	m_2
Final matching ν		b_2, m_3	b_1, b_3, b_4, m_1	m_2
Remaining quota q_c^ν		0	0	0

Table 8: Student b_3 ensures matching ν using the DA with gaps in Example 2

4 Guaranteeing a Stable Matching in a Weighted College Admissions Problem

We want to conclude our analysis by discussing how stability can be guaranteed. As a starting point consider the following situation. All students propose to the colleges simultaneously according to their preferences starting with the most preferred college. As long as each college has enough remaining quota, meaning the sum of the weights of all proposing students is smaller than the college's quota, a college does not reject any student in the first round of proposals. After the first round no student is left without a match. Obviously, in this case we have found a stable matching.

Lemma 1. *If each college's quota is large enough to accept all students who state it as their first preference independent of their weights, there will always be a stable matching that is found by using the DA.*

If instead the colleges' quotas are very tight, we might also find a stable matching. Imagine a situation where each college can accept at most one student, independent of the weight of the student. In this case, gaps do not play a role as it is never the case that a college has enough remaining quota to accept a student who has been rejected before already. In this case, the problem reduces to the original marriage market as was introduced in Gale & Shapley (1962).

Lemma 2. *If each college's quota and the differences in students' weights are small enough, so that each college is allocated to at most one student independent of the student's weight, the resulting matching μ of the DA is always stable.*

We can use these results to ensure a stable matching in an arbitrary weighted college admissions problem. Similarly to Dean et al. (2006) who show that their newly proposed algorithm finds the job-optimal stable assignment among all minimally congested stable assignments we find that a stable outcome can be found by increasing each college's quota by at most $(p_{\overline{s}} - 1)$. Additionally, we show that it is also possible to decrease the colleges' quotas by this amount.

Theorem 3. *To guarantee a stable outcome in a weighted college admissions problem when none exists in its original formulation, it is sufficient to increase or decrease the quota of each college in the cycle by at most $(p_{\overline{s}} - 1)$.*

Proof. As there is no stable outcome in the original problem, we know that a cycle must occur. We can reduce the problem by removing all students who are tentatively accepted by the same college during the whole course of this cycle. Accordingly, the quotas of the colleges are reduced by the weights of these students. A cycle includes a number of students and colleges, but at most one student with weight $p_{\overline{s}}$. To break a cycle we make sure that the student s with the (in the cycle) highest weight p_s is either never rejected or never accepted in the first place. A cycle also always includes another student with a smaller weight than s, at least $p_{\underline{s}} = 1$, who takes turns with s. Therefore, to ensure that s is always accepted (according to Lemma 1), the college's quota must be increased by $(p_s - 1)$ to ensure that s is accepted. By decreasing the quota by the same amount we ensure that she is rejected. As the student's weight is at most $p_{\overline{s}}$, the biggest possible change in the quota is $(p_{\overline{s}} - 1)$. □

To see how the increase or decrease in quotas works in practice, let us look at a final example (Example 4). This example is somewhat more evolved, including more students than our previous examples, to be able to show the effect of increasing or decreasing the quota in non-trivial cases.

Example 4. *Assume there are three colleges c_1, c_2 and c_3 with quotas $q_1 = 4$, $q_2 = 4$ and $q_3 = 3$ and preferences over students $P(c)$. There are eight students: five Bachelor students b with preferences $P(b)$ and three Master students m with preferences $P(m)$. The Bachelor students b_1, b_2, b_3, b_4 and b_5 are of type $\underline{s}$ and have weight $p_{\underline{s}} = 1$ and the Master students m_1, m_2 and m_3 are of type $\overline{s}$ with weight $p_{\overline{s}} = 1.5$. The preferences of the students and the colleges are as follows:*

$P(b_1) : c_1,\ c_2,\ c_3$	$P(m_1) : c_1,\ c_2,\ c_3$	$P(c_1) : b_2,\ m_1,\ b_3,\ b_1,\ b_4,\ b_5,\ m_2,\ m_3$
$P(b_2) : c_1,\ c_3,\ c_2$	$P(m_2) : c_2,\ c_3,\ c_1$	$P(c_2) : b_4,\ b_5,\ b_1,\ m_2,\ b_3,\ b_2,\ m_3,\ m_1$
$P(b_3) : c_2,\ c_1,\ c_3$	$P(m_3) : c_3,\ c_2,\ c_1$	$P(c_3) : m_3,\ m_2,\ m_1,\ b_5,\ b_4,\ b_3,\ b_2,\ b_1$
$P(b_4) : c_2,\ c_3,\ c_1$		
$P(b_5) : c_2,\ c_1,\ c_3$		

To illustrate how the increase or decrease in quotas works, we first show in Table 9 that the mechanism indeed cycles in its original formulation. As you can see, the cycle does not impact type $\underline{s}$ students b_2, b_4, b_5 as well as type $\overline{s}$ students m_1 and m_3. Each one of them is allocated to the same college for the duration of the cycle, namely (c_1, b_2), (c_1, m_1), (c_2, b_4), (c_2, b_5) and (c_3, m_3). Therefore we can reduce the problem by taking out these college-student pairs and consequently reduce the quotas of the colleges as follows: $q_1^{new} = 1.5$, $q_2^{new} = 2$ and $q_3^{new} = 1.5$. In this new formulation of the problem, where we only consider students b_1, b_3 and m_2 and the new quotas, it is easy to see that we still obtain a cycle. Therefore, we use the two methods introduced in Theorem 3 to obtain a stable matching.

In Table 10 we run the DAG with the reduced example where the new quotas of the colleges are decreased *by $p_{\overline{s}} - 1 = 0.5$ each. This means, for example, that the remaining quota of c_1 is $q_1^{new,d} = 1$. We find a final allocation without running into a cycle. Using this allocation, the final allocation of the whole problem is then: (c_1, b_2), (c_1, b_3), (c_1, m_1), (c_2, b_1), (c_2, b_4), (c_2, b_5) and (c_3, m_3). It is straightforward to see that this allocation is stable under the decreased*

Round		College		
		c_1	c_2	c_3
1	proposing	b_1, b_2, m_1	b_3, b_4, b_5, m_2	m_3
	μ^{temp}	b_1, b_2, m_1	b_4, b_5, m_2	m_3
2	proposing	b_1, b_2, b_3, m_1	b_4, b_5, m_2	m_3
	μ^{temp}	b_2, b_3, m_1	b_4, b_5, m_2	m_3
3	proposing	b_2, b_3, m_1	b_1, b_4, b_5, m_2	m_3
	μ^{temp}	b_2, b_3, m_1	$b_1, b_4, b_5, \square$	m_3
4	proposing	b_2, b_3^*, m_1	b_1, b_3, b_4, b_5	m_2, m_3
	μ^{temp}	$b_2, m_1, \square$	b_1, b_3, b_4, b_5	m_2, m_3
5	proposing	b_1, b_2, m_1	b_1^*, b_3, b_4, b_5	m_2, m_3
	μ^{temp}	b_1, b_2, m_1	$b_3, b_4, b_5, \square$	m_2, m_3
6	proposing	b_1, b_2, m_1	b_3, b_4, b_5, m_2	m_2^*, m_3
	μ^{temp}	b_1, b_2, m_1	b_4, b_5, m_2	$m_3, \square$
7	proposing	b_1, b_2, b_3, m_1	b_4, b_5, m_2	m_3
	μ^{temp}	b_2, b_3, m_1	b_4, b_5, m_2	m_3

Table 9: The student-proposing DAG from Example 4

quotas ($q_1^d = 3.5$, $q_2^d = 3.5$, $q_3^d = 2.5$). Please note that student m_2 did not receive a place at a college as she was not accepted anywhere.

Round		College		
		c_1	c_2	c_3
1'	proposing	b_1	b_3, m_2	
	μ^{temp}	b_1	m_2	
2'	proposing	b_1, b_3	m_2	
	μ^{temp}	b_3	m_2	
3'	proposing	b_3	b_1, m_2	
	μ^{temp}	b_3	$b_1, \square$	
4'	proposing	b_3^*	b_1, b_3	m_2
	μ^{temp}	b_3	b_1	
5'	proposing	b_3, m_2	b_1	
	μ^{temp}	b_3	b_1	

Table 10: The student-proposing DAG from Example 4 with reduced quotas

In Table 11 we run the DAG with the quotas of the colleges increased *by*

$p_{\overline{s}} - 1 = 0.5$. *This means, for example, that the remaining quota of* c_1 *is* $q_1^{new,i} = 2$ *. We find a final allocation without running into a cycle. Using this allocation, the final allocation of the complete problem is then:* (c_1, b_1), (c_1, b_2), (c_1, m_1), (c_2, b_3), (c_2, b_4), (c_2, b_5), (c_2, m_2) *and* (c_3, m_3).
It is straightforward to see that this allocation is stable under increased quotas ($q_1^i = 4.5$, $q_2^i = 4.5$, $q_3^i = 3.5$)*. Incidentally, it is also the student-optimal stable matching under the new quotas.*

Round		College		
		c_1	c_2	c_3
1”	proposing	b_1	b_3, m_2	
	μ^{temp}	b_1	b_3, m_2	

Table 11: The student-proposing DAG from Example 4 with increased quotas

By increasing or decreasing the colleges’ quotas by a marginal amount we can ensure stability. Coming back to our original motivation of the kindergarten matching, it might not be an option to increase tight quotas. Instead, it might be helpful to actually decrease them. It seems contradictory at first but can actually ensure stability. Please notice that all colleges that are not part of the cycle are unaffected by these changes, keeping the adjustments as small as possible.

5 Conclusion

There are a number of many-to-one matching markets in which not all participants are weighted equally, such as in the allocation of children to daycare centers, where the weight of a child often depends on its age. However, there is a surprising lack of research on the impact of this heterogeneity on the (weighted) stability of resulting matchings.

In this paper, we analyze a college admissions problem in which there are two or more groups of students who are weighted differently. We find that stability can no longer be guaranteed. Furthermore, the DA, otherwise the preferred matching mechanism in standard many-to-one matching situations from a theoretical point of view, is not useful anymore in finding stable outcomes if they exist. To overcome this problem, we propose an adjusted version of the DA, the DA with gaps. We show that this mechanism reaches stable allocations if they exist and cycles otherwise. Furthermore, by increasing or decreasing the colleges' quotas we can ensure stability. As a drawback, the DAG lacks strategy-proofness and might offer students incentives to misrepresent their preferences. Consequently, policy recommendations for settings with differently weighted participants should not simply call for an implementation of the DA. Instead, the type of situation needs to be analyzed carefully and the mechanism adjusted accordingly.

Acknowledgements

We thank Burkhard Monien for the critical and helpful comments and discussions. We also thank the participants of the 14th and 15th European Meeting on Game Theory, the 14th Meeting of the Society for Social Choice and Welfare and the internal seminar TEAM 2019 at Paderborn University for helpful and encouraging comments. This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Centre 901 "On-The-Fly Computing" (SFB 901) under the project number 160364472-SFB901.

6 Appendix: The college-proposing DAG

Until now we have introduced the DAG as a student-proposing algorithm. In this case, mainly the students' preferences are of interest. This is in line with the literature on school choice where schools or colleges are treated as objects and only the students are considered as agents (Abdulkadiroğlu & Sönmez, 2003). In contrast, the NRMP, the clearing house to match residents or medical interns to hospitals, used a hospital- or in our notion college-proposing algorithm for several decades (Roth, 1984) to ensure that the hospitals receive suitable candidates for their job positions. So, is it possible to focus on the colleges' side with the DAG? The answer is yes. The college-proposing DAG works quite similarly to the student-proposing version. The only differences are the following. Instead of the students, now the colleges propose to as many students as possible according to their preference lists. If a student with a relatively small weight rejects an offer, this might lead to the situation that a college is now longer willing to propose to a certain other student as it now has enough capacity left to propose to a more preferred student with a bigger weight which was not possible before due to capacity constraints. In this case, a gap occurs as a student who was tentatively accepted before is suddenly no longer matched at all. This gap is marked. In every round, at most one marked student is triggered and the already rejected colleges may propose again.

Example 3 (continued). *By using the college-proposing DAG the only existing stable matching is found very quickly, as c_1 is proposing to b_1, b_2, and b_3 while c_2 is proposing to m_2. As no college is rejected, the stable matching is already found.*

To illustrate how gaps may occur and what happens if there is no stable outcome let us look at two other examples.

Example 5. *Given are two colleges, c_1 and c_2, with quotas $q_1 = q_2 = 2$ and preferences over students $P(c)$. There are three students: two Bachelor students b with preferences $P(b)$ and one Master student m_1 with preferences $P(m_1)$. The Bachelor students b_1 and b_2 are of type $\underline{s}$ and have weight $l = 1$ and the Master student m_1 is of type $\overline{s}$ with weight $h = 2$. The preferences of the students and the colleges are as follows:*

$$
\begin{array}{lll}
P(b_1): c_2, c_1 & P(m_1): c_1, c_2 & P(c_1): b_1, m_1, b_2 \\
P(b_2): c_1, c_2 & & P(c_2): b_1, b_2, m_1
\end{array}
$$

With the help of the college-proposing DAG we find a stable outcome which can be seen in Table 12. Please note that there is only one stable matching, which can also be found by using the student-proposing DAG without any occurring gaps.

Round		Student		
		b_1	b_2	m_1
1	proposing	$c_{1.1}$, $c_{2.1}$	$c_{1.2}$, $c_{2.2}$	
	μ^{temp}	$c_{2.1}$	$c_{1.2}$	
2	proposing	$c_{2.1}$	$c^*_{1.2}$	$c_{1.1}$
	μ^{temp}	$c_{2.1}$	$\square$	$c_{1.1}$
3	proposing	$c_{2.1}$	$c_{2.2}$	$c_{1.1}$
	μ^{temp}	$c_{2.1}$	$c_{2.2}$	$c_{1.1}$
Final matching μ		c_2	c_2	c_1

Table 12: Assignment of colleges to students in Example 5 using the DA with Gaps

Example 1 (continued). *From the previous analysis of Example 1 we already know that this example has no stable matching. In Table 13, we illustrate how the college-proposing DAG works in this case. We do not find a final matching here, instead the mechanism cycles. The beginning and end of the cycle is marked in gray. There is no chance to change the order of the triggered marked colleges to escape the cycle either.*

It is easy to show that the results from the previous analysis carry over to the college-proposing DAG by just exchanging the roles of students and colleges.

Round		College		
		b_1	b_2	m_1
1	proposing	$c_{2.1}$	$c_{1.1}$, $c_{2.2}$	$c_{3.1}$
	μ^{temp}	$c_{2.1}$	$c_{2.2}$	$c_{3.1}$
2	proposing	$c_{1.1}$, $c_{2.1}$	$c_{2.2}$	$c_{3.1}$
	μ^{temp}	$c_{1.1}$	$c_{2.2}$	$c_{3.1}$
3	proposing	$c_{1.1}$	$c^*_{2.2}$	$c_{2.1}$, $c_{3.1}$
	μ^{temp}	$c_{1.1}$	$\square$	$c_{2.1}$
4	proposing	$c^*_{1.1}$	$c_{1.1}$, $c_{3.1}$	$c_{2.1}$
	μ^{temp}	$\square$	$c_{1.1}$	$c_{2.1}$
5	proposing	$c_{2.1}$, $c_{3.1}$	$c_{1.1}$, $c_{2.2}$	$c^*_{2.1}$
	μ^{temp}	$c_{2.1}$	$c_{2.2}$	$\square$
6	proposing	$c_{1.1}$, $c_{2.1}$	$c_{2.2}$	$c_{3.1}$
	μ^{temp}	$c_{1.1}$	$c_{2.2}$	$c_{3.1}$

Table 13: The college-proposing DAG cycles when there is no stable matching (Example 1)

Bibliography

Abdulkadiroğlu, A. & Sönmez, T. (2003). School choice: A mechanism design approach. *The American Economic Review*, 93(3), 729–747.

Abizada, A. (2016). Stability and incentives for college admissions with budget constraints. *Theoretical Economics*, 11(2), 735–756.

Biró, P., Fleiner, T., Irving, R. W., & Manlove, D. F. (2010). The college admissions problem with lower and common quotas. *Theoretical Computer Science*, 411(34-36), 3136–3153.

Biró, P., Manlove, D. F., & McBride, I. (2014). The hospitals / residents problem with couples: Complexity and integer programming models. In J. Gudmundsson & J. Katajainen (Eds.), *Experimental Algorithms* (pp. 10–21). Cham: Springer International Publishing.

Biró, P. & McDermid, E. (2014). Matching with sizes (or scheduling with processing set restrictions). *Discrete Applied Mathematics*, 164, 61–67.

Carlsson, S. & Thomsen, S. L. (2015). Improving the allocation of spots in child care facilities for toddlers in germany: A mechanism design approach. *IZA Discussion Paper*.

Dean, B. C., Goemans, M. X., & Immorlica, N. (2006). The unsplittable stable marriage problem. In G. Navarro, L. Bertossi, & Y. Kohayakwa (Eds.), *International Federation for Information Processing, Volume 209, Fourth IFIP International Conference on Theoretical Computer Science-TCS 2006* (pp. 65–75).

Delacrétaz, D. (2019). Stability in matching markets with sizes. unpublished.

Delacrétaz, D., Kominers, S. D., & Teytelboym, A. (2016). *Refugee resettlement*. Technical report, University of Oxford Department of Economics Working Paper.

Echenique, F. & Yenmez, M. B. (2007). A solution to matching with preferences over colleagues. *Games and Economic Behavior*, 59(1), 46 – 71.

Erdil, A. & Ergin, H. (2008). What's the matter with tie-breaking? improving efficiency in school choice. *American Economic Review*, 98(3), 669–89.

European Commission/EACEA/Eurydice/Eurostat (2014). Key data on early childhood education and care in europe. 2014 Edition. In *Eurydice and Eurostat Report*. Luxembourg: Publications Office of the European Union.

Gale, D. & Shapley, L. S. (1962). College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1), 9–15.

Haeringer, G. & Klijn, F. (2009). Constrained school choice. *Journal of Economic Theory*, 144(5), 1921 – 1947.

Hoyer, B. & Stroh-Maraun, N. (2020). Matching strategies of heterogeneous agents under incomplete information in a university clearinghouse. *Games and Economic Behavior*, 121, 453–481.

Kamada, Y. & Kojima, F. (2019). Fair matching under constraints: Theory and applications. Working Paper.

Klaus, B., Klijn, F., & Massó, J. (2007). Some things couples always wanted to know about stable matchings (but were afraid to ask). *Review of Economic Design*, 11(3), 175–184.

Kojima, F. (2015). Finding all stable matchings with couples. *Journal of Dynamics & Games*, 2(3&4), 321–330.

Kojima, F., Pathak, P. A., & Roth, A. E. (2013). Matching with couples: Stability and incentives in large markets. *The Quarterly Journal of Economics*, 128(4), 1585–1632.

McDermid, E. J. & Manlove, D. F. (2010). Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization*, 19(3), 279–303.

Nguyen, T. & Vohra, R. (2018). Near-feasible stable matchings with couples. *American Economic Review*, 108(11), 3154–69.

Pathak, P. A. & Sönmez, T. (2013). School admissions reform in chicago and england: Comparing mechanisms by their vulnerability to manipulation. *The American Economic Review*, 103(1), 80–106.

Roth, A. E. (1984). The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, 92(6), 991–1016.

Roth, A. E. (1985). The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory*, 36(2), 277–288.

Roth, A. E. & Rothblum, U. G. (1999). Truncation strategies in matching markets - in search of advice for participants. *Econometrica*, 67(1), 21–43.

Roth, A. E. & Sotomayor, M. A. (1990). *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monographs. Cambridge University Press.

Veski, A., Biró, P., Poder, K., & Lauri, T. (2017). Efficiency and fair access in kindergarten allocation policy design. *Journal of Mechanism and Institutional Design*, 2(1), 57–104.