Fair Allocation with Exact Capacity Constraints

T.C.A. Madhav Raghavan*

31 July, 2014

Abstract

We consider situations where heterogenous indivisible objects are to be distributed among a set of claimants based on preferences and priorities. We impose the additional restriction that each object has an exact capacity constraint, such that each object is assigned either to a pre-specified (fixed and common) number of agents, or it is not assigned at all. We demonstrate that the well-known incompatibility between fairness and Pareto efficiency in one-sided matching models persists in this model too. We propose a rule which we call the Deferred Acceptance with Improvements (DAI) rule, which is fair and constrained efficient. We also identify a Pareto improvement procedure that always leads us to a fair and constrained efficient allocation in one iteration. We show, however, that the DAI rule is not strategy-proof.

1 Introduction

A premier engineering college in New Delhi, India, offers as a part of its course structure a number of optional courses or 'electives' from the social science department. Every student, in addition to his or her core science and engineering courses, must complete over the five years of the program a certain number of such electives. The social science department offers a variety of courses, from history to political science to philosophy, even economics.

However, as facilities and faculty are limited, so are the total number of courses offered in any particular term. Thus the college stipulates that a student may opt for only one elective in a term, so that there are enough seats in a term for everybody who desires one.

It is not feasible to offer a course if there are few takers for it, because of the cost of time and facilities. The college wishes to offer only those courses that are enough in demand. Thus there is in effect a minimum capacity constraint operating on a course, such that the college finds it infeasible to offer the course to fewer students than that. Also, since classroom or laboratory sizes are limited, there is a maximum number of students that each course can accommodate.

The college wishes to give priority for these electives to students in their second and third years of the program. It feels this is an appropriate time to finish the elective requirements, as the last two years are typically very intensive in the chosen specialisations. Within this broad preference for students by year, the college gives priority to students with a higher GPA, and also sometimes to a subject-specific progression of prerequisites. Thus, for example, a second year student with a 3.6 GPA who has done elementary microeconomics could be given priority for intermediate microeconomics over a fourth year student with a 3.8 GPA.

Before the start of each term, the college asks students to submit their preferences over some potential set of courses to be offered in that term. Based on this information, plus the capacities and the priorities of the various courses, it must prescribe the course allocation for that term. How should it do so?

In this paper we model such a situation. In particular, we look at the case where a college wishes to offer a selection of courses to its students and requires that each student sign up for exactly one of these

*Senior Research Fellow, Indian Statistical Institute, New Delhi. Thesis advisor: Arunava Sen.

courses. In turn, each course has a minimum and maximum capacity and can admit students only within those capacities. For the sake of this paper we make the further assumption that the minimum and maximum capacities for all courses are equal, and the same. That is, for example, each course may admit only exactly twenty-five students, say. While this may seem like an overly restrictive assumption, we believe this is a natural starting point. Relaxing this assumption will not materially change our results, though it will complicate the process.

In keeping with our motivating example, we allow for the fact that a course need not be assigned at all.[1] There are more courses available than may be feasibly assigned together, so in effect the college must determine the solution to a two-part problem: not only must the college decide which selection of courses from the total will be offered, but also which students will be assigned which course.

Course allocations are made on the basis of preference and priorities. Each student has as his or her private information a strict ranking over the available courses, which we call a preference ordering, or simply a preference. This information must be elicited by the college and in general we may wish to award students their preferred courses, as far as possible. On the other hand, each course has a strict ranking over the students, which we call a priority. In contrast to preferences, this priority information is commonly known and fixed, and may be determined by transparent criteria such as GPA, prerequisites, and so on. Priority information captures which students are more eligible for which courses.

So an allocation problem for a college is a collection of students, courses, capacities, course priorities and student preferences. It must use this information to produce a feasible allocation - one in which each student is assigned a course, and every course is assigned to its exact capacity (or to no one). In effect, since the first four are commonly known, the problem becomes one of producing a feasible allocation for any combination of elicited student preferences.

The college wishes that the allocation should satisfy some desirable properties. The first class of properties has to do with fairness. An allocation is deemed unfair for some student if there is a course that she prefers and there is another less eligible student who is assigned that course instead of her. The former student then can be said to have a case of justifiable envy towards the latter. A fair process will avoid this possibility.

The second class of properties has to do with efficiency. An efficient process eliminates waste. In particular, an allocation is inefficient if there is another allocation in which each student receives a course she likes as much, and some student receives a course she strictly prefers. In this case we say that the latter allocation 'Pareto dominates' the former.

However, it has been well documented in the literature that fairness and efficiency are incompatible in the most general environments. We demonstrate by example that this incompatibility persists even in our model with exact capacity constraints.

In the model without minimum constraints, a weaker version of efficiency, called constrained efficiency, is compatible with fairness. A constrained efficient allocation is efficient *within the set* of feasible and fair assignments.[2] The classical Gale-Shapley Deferred Acceptance Rule for instance is simultaneously fair and constrained efficient. We shall describe this rule next. It will provide the starting point for our model.

1.1 The Deferred Acceptance Rule

In the general model with no minimum capacity constraints, the Deferred Acceptance (DA) rule is fair and constrained efficient. It works in a series of rounds, as follows.

In the first round, all students apply to their most preferred course. Any course that receives more applications than its maximum capacity is forced to reject the excess students, provisionally accepting the rest. The students that are rejected are those that are the lowest in that course's priority among its pool of applicants. In the next round, all rejected students apply to their next preferred course that has not rejected them already. A course considers its existing applications plus any fresh ones it might receive, and

[1] This is a departure from other allocation models with minimum constraints. In these papers, each object *must* be assigned up to its minimum constraint.

[2] If some allocation Pareto dominates a constrained efficient allocation, then it must either be infeasible or unfair to some student.

provisionally accepts the top students according to its priority, rejecting the lowest ones that are excess to capacity. If any student is rejected, we go to the next round. The rule terminates in any round in which no student is rejected. All provisional acceptances become final.

Consider the following example. There are three agents $\{1, 2, 3\}$ and three objects $\{a, b, c\}$. Preference and priority information are given in the table below:

Priorities			Preferences		
a	b	c	1	2	3
1	2	3	b	a	a
2	3	1	a	c	b
3	1	2	c	b	c

In the first round, agent 1 applies to her top-ranked object b while agents 2 and 3 apply to a. Since a has two applicants, it rejects the lower-ranked one according to its priority, which is agent 3. Agents 1 and 2 are tentatively assigned b and a respectively. In the next round, agent 3 applies to her next preferred object, which is b. Now b has two applicants, so it rejects the lower-ranked one according to its priority, which is agent 1. Agents 2 and 3 are tentatively assigned a and b respectively. In the next round, agent 1 applies to his next preferred object, which is a. Again, object a must reject one application, and so agent 2 is rejected. In the final round, agent 2 applies to c. There are no more rejections and the rule terminates here, giving us the final allocation $((1, a), (2, c), (3, b))$.

The DA rule is not directly applicable to models with minimum constraints without requiring extra information. We shall discuss the nature of this extra information in more detail and why it is necessary. However, we can find a modified DA rule that is simultaneously fair and constrained efficient. This rule requires no extra information. We call this rule the DA rule with improvements (DAI).

The DAI rule works as follows. We first exogenously select a set of courses that can exactly accommodate all students, and run the conventional DA rule on the restricted environment with only these courses. By the properties of the DA rule, such an allocation will be fair as well as internally constrained efficient [3]. However, our initial selection of courses is arbitrary. There is the possibility that there exists some other allocation, with a *different* selection of courses, that Pareto dominates this one. Our main contribution is to provide a procedure to identify these Pareto improvements whenever they exist. Moreover, we show how we can carry out these efficiency improvements in a manner that preserves the fairness of the original allocation. The composite process is therefore fair and constrained efficient. We also show that the DAI rule needs only one iteration to produce a constrained efficient assignment from any initial selection of courses.

Next we come to questions of strategy. A third property that we would like our rule to satisfy is non-manipulability on the part of students. Since student preferences are private information, the college would like to ensure that students are incentivised to report their true preferences. This is done by eliminating the possibility of undue gain that a student may make by unilaterally reporting false preferences. This is called a profitable manipulation. A rule that can ensure that there is no profitable manipulation for any student is called strategy-proof. It has the additional pleasant hallmark of informational simplicity. Students need only consider what they truly prefer, and need not worry about other students' strategies. The DAI rule is not strategy-proof. However, we show that the opportunities for manipulation on the part of students are limited.

The paper is organised as follows. Section 2 provides a literature review on the DA rule and its properties. We also talk about fair assignments and Pareto improvement procedures in other contexts. Section 3 is an informal discussion on minimum capacity constraints, our rule, and the justification of our procedure. Section 4 begins the formal section of the paper by providing the notation that we use throughout. Section 5 discusses the axioms of fairness and efficiency. Here we show the key incompatibility between the two properties. Section 6 outlines our Pareto improvement procedure and presents our existence result. Section 7 lays out the DAI rule. Section 8 discusses its strategic aspects. Section 9 concludes. All proofs are relegated to the appendix.

[3] An allocation is internally constrained efficient if it is constrained efficient *for that particular selection* of courses. It need not be constrained efficient in general.

2 The Literature

In the context of object allocation, it is well known that Pareto efficiency and fairness are incompatible. There are usually two approaches followed in the literature.

On efficiency, most papers are typically generalisations of the famous top trading cycles mechanism (TTCM) attributed to David Gale (Shapley and Scarf (1974)), and include inheritance rules (Pápai (2000), Pycia and Ünver (2013)), and sequential priority rules (Svensson (1999), Hatfield (2009), Pápai (2001) etc.) Such rules are Pareto efficient and strategy-proof but not fair. Since we are interested in fairness, we do not pursue this line of research here.

Fairness is the one-sided analogue property to stability in the two-sided matching environment[4] (see for example Abdulkadiroğlu and Sönmez (2003)). In this literature, there are several key results. The famous Gale-Shapley agent-optimal stable mechanism (AOSM)(Gale and Shapley (1962)) always produces a stable matching in the two-sided case. Its direct adaptation to the one-sided model (called the deferred acceptance (DA) rule, described above) always produces an envy-free or fair matching. Moreover, this mechanism Pareto-dominates any other mechanism that is stable/fair (Gale and Shapley (1962)). Also, the DA rule is strategy-proof (Roth (1982b), Dubins and Freedman (1984)). In fact, the DA rule is the only mechanism that is individually rational, fair, non-wasteful, and strategy-proof (Alcalde and Barberà (1994)). There are other characterisations of the DA rule as well (see for example Kojima and Manea (2010)).

In the one-sided model, such as ours, object 'preferences' are interpreted as priorities. Under certain conditions on these priorities, the DA rule and the TTCM can be shown to be equivalent. In particular, Kesten (2006) shows the equivalence of the DA rule and the TTCM when priorities are acyclical, i.e., satisfy a very particular restriction. For more on acyclicity, see also Ergin (2002).

Some recent papers have introduced the problem of minimum constraints to the allocation model. In a closely related paper, Fragiadakis et al. (2012) examine a model where each object has a minimum capacity that must be satisfied. They introduce adaptations of the DA procedure that satisfy combinations of properties. In particular, their ESDA procedure is strategy-proof and satisfies a weak version of fairness and non-wastefulness, while the MSDA procedure satisfies an even weaker notion of fairness, but is non-wasteful and strategy-proof. However, in their model they assume that all objects *must* be assigned, thus all minimum capacities must be satisfied. This is in contrast to our model, where some objects may not be assigned at all. Thus our results are qualitatively different from theirs.

In another related model, Ehlers et al. (2011) and Abdulkadiroğlu and Ehlers (2006) look at the case where agents may be classified according to type, and each object has a minimum capacity for each student type, which must be satisfied. They too show that there might not exist assignments that satisfy standard fairness and non-wastefulness properties, whereas constrained non-wasteful assignments which are fair for same type students always exist. They introduce a 'controlled' version of the (DA) rule with an improvement stage that finds a Pareto optimal assignment among such assignments.

In the context of Pareto improvements, Erdil and Ergin (2008) show how to go from any stable assignment to the student-optimal stable assignment when indifferences may exist in priorities. Indifferences are broken according to an arbitrary tie-breaker, and then the DA rule is used to get a stable assignment. However, because of the arbitrariness of the tie-breaker, and the presence of indifferences, this assignment may not be a Pareto-efficient stable assignment. But the repeated use of what they call the stable-improvement-cycles technique leads inevitably to a Pareto-efficient stable assignment, if one exists. Their key result, similar in spirit to ours, is that if an assignment is Pareto dominated by another stable assignment, then there must exist a stable-improvement-cycle. However, their model considers indifferences and not minimum capacities, and is thus substantially different from our model here.

3 An Informal Discussion

Suppose there are M courses and N students. We assume that the capacity of each course is the same (and equal to q) in order to not discriminate between different combinations of courses, i.e., so that every

[4]Two-sided matching, such as marriage markets, involves private preferences on both sides of the market.

combination of courses is equally likely to appear in a feasible allocation. The total available seats in all the courses is then qM.

If $qM < N$, then no feasible allocation exists, as there are not enough seats for each student. If $qM = N$, then minimum capacities become irrelevant, as there are only as many seats as students. Thus if we run the conventional DA rule as described in the introduction with maximum capacities set as q for each course, we will always get a feasible allocation. Moreover, this allocation will be fair, constrained efficient, and the rule will be strategy-proof.

However, consider the case where $qM > N$. There are more seats available than students to fill them. In particular, let $qm = N$ for some $m < M$. A feasible allocation will thus contain some *selection of* m *courses* from the whole set, since not all can be simultaneously offered. So now we have a two part problem in which we have to determine not only the subset of m courses that will be offered, but also the allocation of q students to each course.

Once we have the courses selected, we can run the regular DA rule to get a fair and constrained efficient allocation. So let us look at the course selection problem first. There are at least three approaches we could take. We could:

1. Systematically add courses to our collection until we reach m
2. Systematically eliminate courses from the available set until we are left with m
3. Start with an arbitrary selection of m courses and progressively exchange courses until we have the 'right' combination.

We consider each of these approaches in turn.

3.1 Course Addition

Assume we have to select the required number of courses somehow. A natural starting point would be the sequential priority rule, in which students are ordered according to some ranking, and successively pick their desired courses until the requisite number is reached. In such a scenario, the remaining students would then be assigned one or other of these selected courses. However, this approach suffers from a few drawbacks.

Firstly, consider the ordering of students that determines the sequence in which they pick the courses. This assumption has some intuitive justification in the context of course selection (since students can be ranked according to their CGPA), and has parallels in other matching problems. In the US military, for example, cadets are ranked according to a single merit list based on overall performance (Sönmez and Switzer (2011)). In school choice problems, in most countries students are required to take a common exam, which serves as the basis for admissions to many schools (Abizada and Chen (2011)). Fragiadakis et al. (2012) use an overall ranking, which they call a 'master list', as a key ingredient in their models. However, note that this ranking is an extra piece of information that is required, over and above preferences and course priorities.

Secondly, we may not always be able to respect the selection of earlier students in the sequential priority. This is because the sequential priority ranking may be different from how a course ranks students in its own course priority. So there may be subsequent students in the sequential priority who are higher-ranked in a course's priority than the student who originally selected it. In this case we cannot continue to assign it to the student who selected it without violating fairness.

To see this, consider the following example: Suppose there are four students $\{1, 2, 3, 4\}$ and three courses $\{a, b, c\}$, and each course has a capacity of 2. Suppose the sequential priority is $(1, 2, 3, 4)$, students $1, 3, 4$ most prefer course a, student 2 prefers course b, and the priority for course a is $(3, 4, 1, 2)$. If we follow the sequential priority rule, student 1 goes first and will pick course a. Then student 2 will pick course b. Now if student 1's choice is to be respected, then only one of students $3, 4$ can be assigned course a. But whichever one of them does not get assigned a will feel justified envy towards student 1. Thus this allocation will not be fair.

Even with the sequential priority rule, the selection of a course and its allocation to the selecting student can only be provisional. But this gives rise to the possibility of manipulation. A student higher in the sequential priority may deliberately pick a course in order to affect the choices of later students. In turn, this might grant her a preferred course. Such a rule may not be strategy-proof.

3.2 Course Elimination

Suppose instead that we start with the entire set of courses, looking for a way to eliminate undesired courses. Picture the DA rule, and suppose that in the first round students apply to their preferred course among the entire set on offer. The provisional allocation in this round typically will be infeasible. Infeasibility in this context comes in two flavours: (1) There may be one or more courses that receive more applications than their capacities; and (2) There may be one or more courses that receive less than their capacities.

The DA rule tells us what to do in the first case, i.e., of *oversubscription*. We simply appeal to the relevant course's priority, and reject the application of the lowest-ranked students that are excess to capacity. But the second possibility (which we call *undersubscription*) is more problematic. What if there is more than one course that receives less applications than it requires? Do we eliminate only one of them? If so, which one? Alternatively, do we reject the applications of some of those students? If so, which ones?

Just as in the earlier case, we could resort to extra information. We could assume a ranking over courses, such that we eliminate the lowest-ranked undersubscribed course and reassign its applicants. However, this introduces a discrimination between courses. It may result in the same set of courses being offered in most allocations. Plus, we would like to avoid a situation where a History course is always favoured over a Political Science course, say.

Instead, we could assume a ranking over students, such that an undersubscribed course that receives applications from higher-ranked students is retained in favour of another that has only lower-ranked applicants. However, consider the following scenario: students who have applied to course a are the highest-ranked in its priority, but lowest-ranked in the overall ranking. Course a is undersubscribed and so these students are rejected. Course a is eliminated as a result. These students apply to course b. They are higher ranked in course b's priority than some of the existing applicants, who are now in turn rejected. These students prefer course a to any other course. But course a is now eliminated, so either they cannot apply to it, or doing so violates fairness in terms of the original rejected students. There are also significant efficiency losses in this scenario.

We also cannot let the choice of course to be eliminated depend on student preferences without running the risk of manipulation.

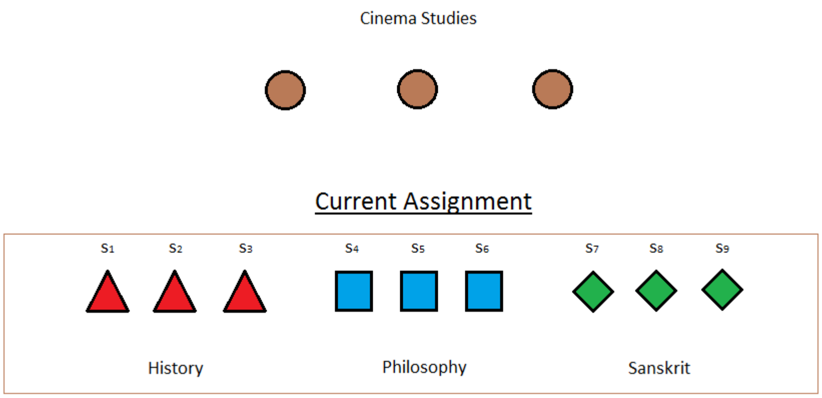
3.3 Our Model

In this paper we take the third approach. We design a rule that first makes an allocation with some arbitrary selection of courses, and then consider efficiency improvements.

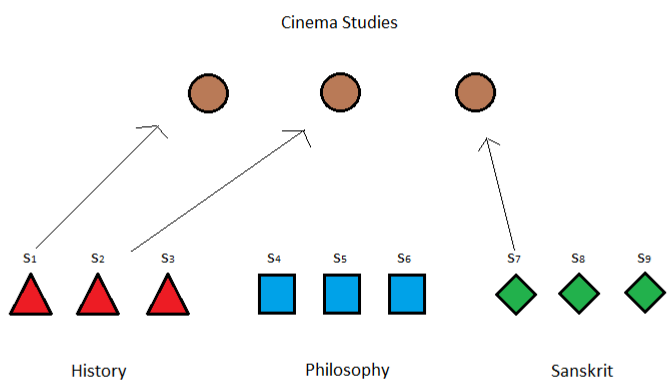
So first we arbitrarily select m courses that can exactly accommodate all students, i.e, such that $qm = N$. We have left out some courses at this stage. We then run the classical DA rule on the restricted environment with just this selection of courses.

By the properties of the DA rule, we are guaranteed an allocation that is fair and internally constrained efficient. As mentioned earlier, internal constrained efficiency means that there is no other fair allocation *with the same selection of courses* that Pareto dominates this one. But this allocation may not be constrained efficient in general. In particular, since the initial selection of courses was exogenous, and did not depend on the preference profile, it might be that there is an unassigned course that some students prefer to their current allocation.

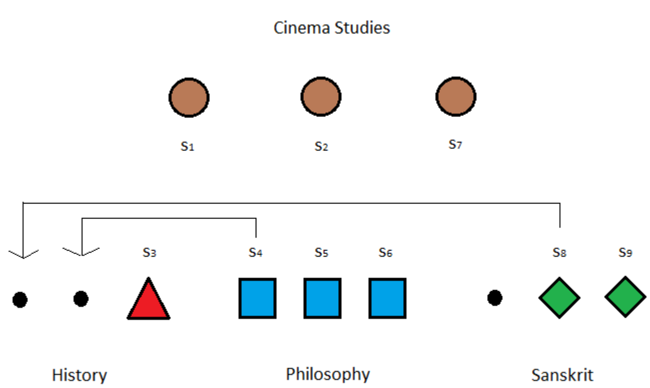
So consider the following scenario where students s_1 to s_9 are assigned courses in History, Philosophy and Sanskrit in a fair and internally constrained efficient manner. There are three seats in each course. But suppose that there is additionally a course in cinema studies which could be offered to them.



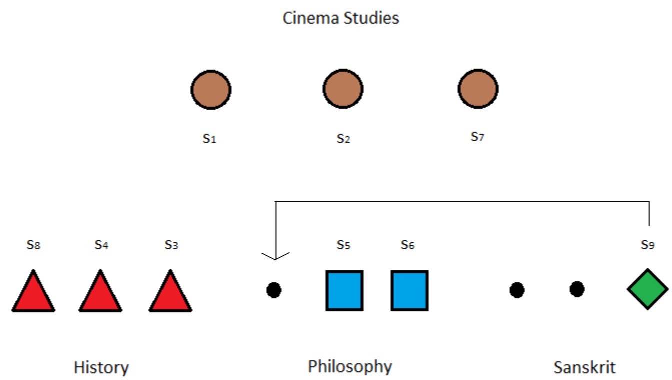
If there are enough claimants for the Cinema Studies course, i.e., students who desire the course at the current allocation, it may be possible to transfer the top claimants to the course (see figure below).



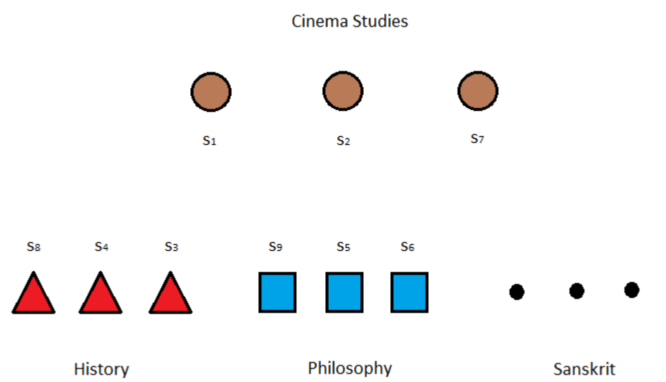
But doing so will leave vacancies in the courses they leave behind. Thus this provisional allocation is not feasible, though it is fair and also a Pareto improvement on the original allocation. So next we see if we can fill the slots that are left behind. Note that at each stage if we are able to fill a slot we must do so by moving the *top claimant of that slot* according to the priority of the relevant course. This is essential to ensure fairness of the overall allocation. Suppose we can do so as below.



This will leave new vacancies, which we fill again, if possible.



Suppose the result is an allocation as in the figure below.



By following this sequential improvement procedure, we have achieved not only a feasible allocation, but one that is also fair and a Pareto improvement on the original allocation.

In essence we have built a Pareto improvement chain for every slot in the new course, such that each chain terminates in a slot in some other course. The course where the chains terminate can be removed and replaced with the new course. Of course, the scenario can be more complicated, in that the chains may be interlinked with those for another course. But the key insight is that if at any point if we can find a collection of chains, one for each slot in a course, such that all chains terminate in slots in the same courses, then we can add the new courses and delete the old ones. This will give us a Pareto improving allocations that is also fair and feasible. In particular, the original allocation is not constrained efficient.

In fact, our first theorem characterises precisely this situation. It shows that the original allocation is internally constrained efficient but not constrained efficient if and only if such a collection of improvements exists. That is, there is a sequence of fairness-preserving moves that leads us from every slot in a new course to every slot in some course that will be now dropped. We also outline a procedure to find such improvements.

We thus build a composite rule, the Deferred Acceptance with Improvements (DAI), which works as follows. We exogenously select some courses, run the DA rule to get an allocation, and check it for improvements. If improvements exist, and if there is more than one possible improvement, then we select[5] one of them and run the DA rule again for the new selection of courses. This resulting allocation will be constrained efficient and fair. We also show that a constrained efficient allocation can be found by just running this entire procedure once.

[5]The selection of improvements is a non-trivial exercise, and we shall discuss it in greater detail in a later section.

We then test the DAI rule for strategic implications. The DAI rule is not strategy-proof. But the possibilities of manipulation are limited. We show that no student can induce the favourable addition of a course or favourably change her assignment within the set of assigned courses via a manipulation. The reason the rule is not strategy-proof is that a student can cause a potentially Pareto improving course to be removed from contention in favour of another course that is also in contention and that she prefers. We demonstrate by example and show that this is the only way a student can manipulate.

A key benefit of the DAI rule, above other solutions to minimum capacity constraints, is that it requires no additional information (i.e., no additional rankings over courses or students). In the sections that follow we formally demonstrate the results and arguments above.

4 Notation and Definitions

- There is a finite set of *students* $\mathcal{S} = \{s, t, u, ...\}$, and a finite set of *courses* $\mathcal{C} = \{c, d, e, ...\}$.
- Each course $c \in \mathcal{C}$ has an *exact capacity* of q. We assume that $|\mathcal{S}| = qm$ for some integer m and that $|\mathcal{C}| > m$.
- An *allocation* is a mapping $\mu : S \cup C \to 2^{\mathcal{S} \cup \mathcal{C}}$ such that:
    1. $\mu(s) \in C$ for all $S \in \mathcal{S}$.
    2. $\mu(c) \subseteq S$ for all $c \in \mathcal{C}$.
    3. $\mu(s) = c$ if and only if $s \in \mu(c)$.

  In an allocation, every student is mapped to a course, every course is mapped to a set of students, and a student is mapped to a course if and only if she belongs to the set of students that the course in turn maps to. For an allocation μ, a student s and a course c, $\mu(s)$ and $\mu(c)$ refer to student s and course c's *assignment* in μ, respectively.
- An allocation μ is *feasible* if $|\mu(c)| \in \{0, q\}$ for all $c \in \mathcal{C}$, and $|\mu(s)| = 1$ for all $s \in \mathcal{S}$. An allocation is feasible if every course is assigned either to no student, or to exactly q students, and every student is assigned to exactly one course. Let $\mathcal{A}$ denote the set of all feasible allocations.
- Preferences over assignments are strict. Formally, student $s \in \mathcal{S}$ has *preferences*, denoted R_s, that are given by a binary relation over $\mathcal{C}$. The binary relation is reflexive (for all c, cR_sc), complete (for all c, d, cR_sd or dR_sc), transitive (for all c, d, e, cR_sd and dR_se imply cR_se) and antisymmetric (for any c, d, cR_sd and dR_sc imply $c = d$). Here cR_sd is interpreted as 'course c is at least as good as course d for student s under preferences R_s'. The associated strict relation is given by P_s, such that cP_sd if cR_sd and $c \neq d$. For any c, d, cP_sd means 'c is preferred by s to d under preferences R_s'. We assume that all courses are acceptable to all students.
- Agent preferences over allocations are selfish, in that they care only about the assignment they receive. Agents are indifferent between all allocations that give them the same assignment. An agent's preferences between two allocations that give her different assignments are governed by her preferences over the respective assignment she receives.
- A collection of preferences for all agents is called a *preference profile*, or simply a *profile*, and is denoted by $R = (R_1, ..., R_N)$. The set of all preference profiles is $\mathcal{R}$. In this model we shall usually suppress reference to $\mathcal{R}$, with the understanding that we operate on the full domain of preferences everywhere. As is the convention, we write R_{-i} for a sub-profile of preferences of all agents other than i. Similarly, for a subset of agents M, we write R_M and R_{-M} to denote the sub-profile of preferences of agents in subsets M and $\mathcal{N} \setminus M$, respectively.

- Each course $c \in \mathcal{C}$ has a strict ranking $\succ_c$ over students, which we call a *priority*. A priority is given by a reflexive, complete, transitive and antisymmetric binary relation over $\mathcal{S}$. For any priority $\succ_c$ and students s and t, $s \succ_c t$ means that 'student s has a higher priority for course c than student t'. We assume that all students are acceptable to all courses. We refer to a collection of priorities for all courses as a priority structure, and denote it as $\succ$. The priority structure is exogenous, fixed, and common knowledge.

- An *allocation problem* is the tuple $(\mathcal{S}, \mathcal{C}, q, \succ, R)$. When $\mathcal{S}, \mathcal{C}, q, \succ$ are fixed, as in what follows, we will refer to an allocation problem simply as R.

- An *allocation rule* (or simply a *rule*) $f : \mathcal{R} \to \mathcal{A}$ solves every allocation problem, i.e., associates every preference profile R with a feasible allocation. For any student s, $f_s(R)$ is the assignment she receives at preference profile R according to the rule f. Similarly, for any subset of students T, $f_T(R)$ is the $|T|$-dimensional vector of assignments of T at R, according to f.

5 Properties of Allocation Rules

The first property we wish to implement is fairness. An allocation is unfair if there is a student who does not get a course she desires, and loses out to some other student who is below her in that course's priority. If that is the case, then this student justifiably envies the other student. We wish to avoid such a situation. Formally:

Definition **1.** For a preference profile R, An allocation μ is *unfair to student s* if there is a course c and a student t such that $cP_s\mu(s)$, $t \in \mu(c)$ and $s \succ_c t$, i.e., if student s prefers student t's assignment (course c) to her own, and also has a higher priority for c than t. An allocation μ is *fair* if it is not unfair to any student. A rule f is *fair* if the allocation $f(R)$ is fair for every preference profile R.

The second class of properties we are interested in has to do with efficiency. In its strong form, we say an allocation is efficient if there is *no other* allocation that Pareto dominates it, i.e., in which all students are at least as well off according to their preferences, and at least one student is strictly better off, in the sense that she is assigned a course that she prefers. Formally:

Definition **2.** For a preference profile R, An allocation μ is *Pareto efficient* if there is no other allocation ψ such that $\psi(s)R_s\mu(s)$ for all $s \in \mathcal{S}$ and $\psi(t)P_t\mu(t)$ for some $t \in \mathcal{S}$. A rule f is *Pareto efficient* if, for all preference profiles R, the allocation $f(R)$ is Pareto efficient.

Pareto efficiency is incompatible with fairness in general. The following example, which we adapt from Roth (1982a), demonstrates this.

Example **1.** Let $s_1, s_2, s_3, s_4, s_5, s_6$ be six students and let a, b, c be three schools with exactly two seats each. The constraint structure, priority structure and preference profile is given in the following table.

$\succ_a$	$\succ_b$	$\succ_c$	P_{s_1}	P_{s_2}	P_{s_3}	P_{s_4}	P_{s_5}	P_{s_6}
s_1	s_3	s_3	b	b	a	a	a	a
s_2	s_4	s_4	$\boxed{a}$	$\boxed{a}$	$\boxed{b}$	$\boxed{b}$	b	b
s_5	s_1	s_1	c	c	c	c	$\boxed{c}$	$\boxed{c}$
s_6	s_2	s_2						
s_3	s_5	s_5						
s_4	s_6	s_6						

It is easy to check that the allocation in boxes is the only fair allocation, which is not Pareto efficient as it is dominated by the allocation in circles. However, the allocation in circles, while Pareto efficient, is not fair as s_5 envies s_4 for course a.

So we weaken our definition of efficiency to constrained efficiency. An allocation is constrained efficient if there is *no other feasible and fair allocation* that Pareto dominates it. Formally:

DEFINITION **3.** An allocation μ is *constrained efficient* if there is no other feasible and fair allocation ψ such that $\psi(s)R_s\mu(s)$ for all s, and $\psi(t)P_t\mu(t)$ for some t.

For an allocation μ, let C_μ denote the *set of courses assigned in* μ, i.e., $C_\mu = \{c \in \mathcal{C} | \mu(s) = c$ for some $s \in \mathcal{S}\}$. Next we define the notion of internal constrained efficiency, which has to do with efficiency with respect to the courses selected in a particular allocation.

DEFINITION **4.** An allocation μ is *internally constrained efficient* if there is no other fair and feasible allocation ψ with $C_\psi = C_\mu$, such that $\psi(s)R_s\mu(s)$ for all s, and $\psi(t)P_t\mu(t)$ for some t.

An allocation that is Pareto efficient is also constrained efficient, and an allocation that is constrained efficient is also internally constrained efficient, but the reverse implications may not be true.

6 Pareto Improvements

In this section we elaborate on a technique that allows us to recover constrained efficiency from an internally constrained efficient allocation. Note that an internally constrained efficient allocation need not be constrained efficient, because we do not consider courses outside of the allocation when evaluating the former. There may be some other course that is not assigned, that enough students prefer, such that we can suitably transfer and reassign students making some of them better off without making anybody else worse off, while preserving fairness. We build the notion of an improvement to capture such situations.

6.1 Pareto Chains

Since each course has an exact capacity q, we say that each course $c \in \mathcal{C}$ has q *slots* $\{o_1^c, o_2^c, ..., o_q^c\}$, denoting the available positions in the course. We denote a generic slot for a course c as o^c.

For an allocation μ and a course c, let $D_\mu(c) = \{s \in \mathcal{S} | cR_s\mu(s)\}$ be the set of students who *desire* c *at* μ. Let D_μ refer to the collection $\{D_\mu(c)\}_{c\in\mathcal{C}}$. We call D_μ the *claimant profile at* μ.

Also, for course c and any subset of students T, let $top_k(\succ_c, T)$ denote the k *top students in* c*'s priority among students in* T. s

First, we define a Pareto chain. Suppose we are given an allocation at a preference profile. Consider some course that is unassigned at this allocation. A Pareto chain is a sequence of distinct students and their assignments such that the first student in the chain prefers this unassigned course, and each subsequent student prefers the assignment of the student immediately before her in the chain. A Pareto chain will be associated with a particular slot in the unassigned course.

DEFINITION **5.** Let μ be an allocation, let R be a preference profile, let $c \notin C_\mu$ be a course, and let o^c be a slot in c. We say that a finite sequence of students and courses $\gamma = \{(s_k, c_k)\}_{k=1}^K$, with $1 \leq k \leq K$, is a *Pareto chain originating at* o^c if:

1. $s_i \neq s_j$ for all $i, j \in \{1, ..., K\}$
2. $s_i \in \mu(c_i)$ for all $i \in \{1, ..., K\}$
3. Set $c_0 \equiv c$. Then $c_{i-1}R_{s_i}c_i$ for all $i \in \{1, ..., K\}$, with strict preferences for at least one student.
4. For each c_{i-1}, $s_i = top(\succ_{c_{i-1}}, D_\mu(c))$.

We call a particular pair (s_k, c_k) a *link* in the chain. The first condition requires that all students in the chain be distinct, while the second condition pairs each student in a link with her corresponding assignment. The third condition highlights the chain quality, by requiring that each student prefer the assignment of the previous student in the chain, with the first student preferring the unassigned course. The fourth condition states that each student in the chain is the top student for the course she desires among those who desire it at that allocation.

We now introduce some notation to simplify exposition. For a chain γ and a particular k, we denote the corresponding s_k as $\gamma^s(k)$ and the corresponding c_k as $\gamma^c(k)$. We denote the last student in the chain, $\gamma^s(K)$, by $\bar{\gamma}^s$, and the last course in the chain, $\gamma^c(K)$, by $\bar{\gamma}^c$. Similarly, we denote the origination course for a chain γ by $\gamma^c(0)$. Denote a collection of chains satisfying the above properties by Γ.

So, for a course c and some slot o^c, if a Pareto chain exists, it looks like this, where the arrows mark the direction of preference, and where $\gamma^c(0) = c$:

$$\begin{array}{c} \gamma^c(0) \\ \hline \uparrow \\ (\gamma^s(1), \gamma^c(1)) \\ \uparrow \\ (\gamma^s(2), \gamma^c(2)) \\ \uparrow \\ \ldots \\ \uparrow \\ (\bar{\gamma}^s, \bar{\gamma}^c) \\ \hline \end{array}$$

A Pareto chain is designed to capture a potential sequence of transfers that would make some students better off, while preserving the assignments of other students, in a fair manner. That is:

Definition **6.** For an allocation μ, a course $c \notin C_\mu$ and a Pareto chain γ associated with some slot $o^c \in c$, we say that an allocation μ' is an *upgrade of* μ *via* γ if $\mu'(\gamma^s(k)) = \mu(\gamma^s(k-1))$ for all k and $\mu'(s) = \mu(s)$ for all other students.

It is clear that an upgrade, if it exists, is a Pareto improving allocation, since at least one student in the chain is strictly better off and all other students are equally well off. It is also fair.[6] We now provide the notion of an *improvement*, which is a collection of distinct Pareto chains that originate and terminate 'coherently'.

Definition **7.** Let μ be an allocation with courses C_μ. Let C be a selection of m courses with $C \neq C_\mu$. A collection of chains Γ^C is an *improvement of* μ if:

1. For any $c \in C \setminus C_\mu$, we have that $|\{\gamma_i \in \Gamma^C : \gamma_i^c(0) = c\}| = q$.
2. For any $c' \in C_\mu \setminus C$, we have that $|\{\gamma_i \in \Gamma^C : \bar{\gamma}_j^c = c'\}| = q$.
3. $\bigcap\limits_{\Gamma^C} \gamma_j^s(k) = \phi$.

The first condition requires there to be a Pareto chain in the collection for every slot in every course that we seek to add. The second condition requires that there be a Pareto chain terminating in every slot in a course that we seek to drop. The third condition requires each student in the collection of chains to be distinct.

We are now ready to state our first theorem. Theorem 1 says that an internally constrained efficient allocation is not constrained efficient when we can find an improvement for some other set of courses such that the chains terminate 'coherently'. In other words, when we perform the upgrade related to the improvement, then the resulting allocation is feasible.

Theorem **1.** *Let R be a profile and let μ be an internally constrained-efficient allocation for some selection of courses C_μ at R. Then, μ is not constrained efficient if and only if there exists a selection of courses $C \neq C_\mu$ such that Γ^C is an improvement of μ.*

[6] It may not be feasible.

The proof is in Appendix A. We provide an informal discussion here.

Theorem 1 characterises the situation where an allocation is internally constrained efficient but not constrained efficient. If this is true, then there must be some other courses such that, when we build the Pareto chains for slots in those courses, the resulting collection of chains is an improvement of the original allocation, and also that the resulting upgrade is feasible.

In one direction, it is clear that if such an improvement exists, the resulting upgrade is feasible, fair, and Pareto dominates the original allocation, and thus the latter cannot be constrained efficient.

To prove the converse, we assume that the allocation is not constrained efficient. Then there must be a constrained efficient allocation which Pareto dominates it. Furthermore, this constrained efficient allocation must differ from the original allocation by at least one course.[7] We build the collection of Pareto chains for the courses in this allocation, and show that the collection is an improvement according to the definition, and that each chain terminates in a course that is not present in the constrained efficient allocation. We build these chains iteratively, identifying the top claimant for a course in each step, and upgrading students' assignments as we go along. The key insight is that at each stage some student is strictly better off, and no student is ever worse off, so eventually we get to the constrained efficient allocation, which is feasible and fair.

Thus we have a procedure that allows us to evaluate the constrained efficiency of an allocation. We use this feature to define our composite rule in the next section.

7 Deferred Acceptance with Improvements

The Deferred Acceptance with Improvements (DAI) rule operates in three stages. Firstly, we select some courses and make assignments following the classical DA procedure, restricting preferences to just those courses. By the properties of the DA rule, such an allocation is fair and internally constrained efficient. We then look for improvements to this allocation. If we find one, we build a new set of courses accordingly. The third stage is a repetition of the first stage for the new selection of courses. The resulting allocation will be fair, feasible and constrained efficient.

Initialisation

Exogenously select a subset of courses $C \subset \mathcal{C}$ such that $|C| = m$. Then $q|C| = |\mathcal{S}|$, i.e., there are just enough courses to exactly accommodate all students.

Assignment Step

Restrict preferences to courses in C, and run the classical Deferred Acceptance procedure restricted to these courses. That is:

1. In the first round, all students apply to their most preferred course in C. Any course c receiving more than q applicants provisionally accepts the applications of the top q students among the applicants according to $\succ_c$, rejecting the rest.

2. In any subsequent round, rejected students apply to their most preferred course in C that has not rejected them already. Each course c evaluates its existing applicants (if any) plus new applicants (if any), provisionally accepting the top q applications according to $\succ_c$, and rejecting the rest. If any course rejects a student, we go to the next round. The procedure stops in a round where there are no more rejections.

It is clear that since $q|C| = |\mathcal{S}|$, that this procedure will terminate in a finite number of steps with each course in C receiving exactly q applicants, since any course receiving $k > q$ applicants in any round must reject $k - q$ applications. Courses outside C receive no applicants. Moreover, each student is assigned to a course in C. Thus the resulting allocation is feasible. By the properties of the classical DA rule, this allocation is fair and internally constrained efficient.

[7] Otherwise the original course could not have been internally constrained efficient.

By Theorem 1, we know that to check if this allocation is constrained efficient, we need only to look for improvements. So we run the Pareto improvement procedure as follows:

Improvement Step

Let μ be the allocation generated by the Assignment Step.

Recall that for an allocation μ and a course c, $D_\mu(c) = \{s \in \mathcal{S} | cR_s\mu(s)\}$ is the set of students who desire c at μ. Similarly, $D_\mu = \{D_\mu(c)\}_{c\in\mathcal{C}}$ is the claimant profile at μ.

Also, for course c and any subset of students T, $top_k(\succ_c, T)$ denotes the k top students in c's priority among students in T.

1. For each $c \notin C_\mu$, we build the Pareto chains for the slots of c as follows:
   (a) Let $S_c = top_q(\succ_c, D_\mu(c))$. If $|S_c| < q$, discard this course. If not:
   (b) Create a chain γ_i for each slot $o_i^c \in c$, with $\gamma_i^s(1) = s_i$ for some distinct $s_i \in S_c$, and let $\gamma_i^c(1) = \mu(s_i)$.
   (c) For every $k > 1$, and every i, let $\gamma_i^s(s_k) = top(\succ_{\gamma_i^c(k-1)}, D_\mu(\gamma_i^c(k-1)) \setminus \{s_{k-1}\})$ if this student exists, and let $\gamma_i^c(k) = \mu(\gamma_i^s(k))$.
   (d) At any stage, if the same student repeats in any two chains, remove her from the chain she less prefers and remove all subsequent students in that chain.
   (e) At any stage, if all chains contain the same course, then remove all additional links and go on to the next course.
   (f) Otherwise repeat from step (c) until no other links can be formed.
   (g) Repeat the above steps for all other courses $c \notin C_\mu$.
2. When all chains are built, search for a collection of subchains Γ such that:
   (a) For any c with a $\gamma \in \Gamma$ such that $\gamma^c(0) = c$, we have that $|\{\gamma_i \in \Gamma : \gamma_i^c(0) = c\}| = q$.
   (b) For any c' with a $\gamma \in \Gamma$ such that $\bar{\gamma}^c = c'$, we have that $|\{\gamma_i \in \Gamma : \bar{\gamma}_i^c = c'\}| = q$.
   (c) $\bigcap_{\Gamma^C} \gamma_{ij}^s(k) = \phi$.

If we find such a collection of subchains (and there may be more than one collection), we *select*[8] one of these collections Γ, generate C' from C by adding the courses corresponding to the first nodes of chains in Γ and deleting the courses corresponding to the final nodes of chains in Γ. Then we repeat the Assignment Step with C'. The resulting allocation will be feasible, fair and constrained efficient.

7.1 Example

Let $\{a, b, c, d, e\}$ be five courses and let $\{1, 2, 3, 4, 5, 6\}$ be six students, and suppose that $q = 2$. The preference and priority information is given in the following table.

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	d	a	d	e	e	b
4	3	5	3	2	a	e	b	d	b	a
5	5	1	6	3	b	d	c	a	a	d
6	2	3	4	5	c	c	a	c	c	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

Suppose we start with the initial course selection $\{a, b, c\}$. Then the DA rule produces the allocation given below in boxes:

[8] We discuss in the next sub-section how to select from the possible collections.

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	d	a	d	e	e	$\boxed{\text{b}}$
4	3	5	3	2	$\boxed{\text{a}}$	e	$\boxed{\text{b}}$	d	b	a
5	5	1	6	3	b	d	c	$\boxed{\text{a}}$	a	d
6	2	3	4	5	c	$\boxed{\text{c}}$	a	c	$\boxed{\text{c}}$	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

Now, to check if the allocation is constrained efficient, we build the potential Pareto chains for unassigned courses $\{d, e\}$.

d		e	
o_1^d	o_2^d	o_1^e	o_2^e
$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
(1,a)	(3,b)	(4,a)	(2,c)
$\uparrow$	$\uparrow$	$\uparrow$	
(2,c)	(5,c)	(5,c)	

Thus we have the choice of selecting either e or d, and dropping c.

7.2 Selecting the Improvement

As can be seen from the example above, there may be several improvements that may arise as a result of the procedure. We have to select one of them. We could of course select one of these improvements arbitrarily, and we would be left with a fair and feasible allocation that is a Pareto improvement of the original. But how we select the improvement will matter in terms of the overall efficiency of the procedure. We claim that it is possible to reach from any arbitrary initial selection of courses to a selection that guarantees a constrained efficient allocation via a single round of the Improvement Step. This requires us to always be able to select a constrained efficient allocation at this stage.[9]

In what follows, we describe how to efficiently select a improvement from all the ones that are available after the Improvement Step.

First we need some notation. Let R be a preference profile and let μ be the allocation produced by the DA rule for some selection of courses C. We call μ the *base allocation at R for C*. Let $\mu^1, ...\mu^k$ be k the allocations resulting from k different improvements generated by the Improvement Step. For each μ^i, let $S(\mu^i) = \{s \in \mathcal{S} : \mu^i(s) \neq \mu(s)\}$ be the set of all students whose assignments change from μ to μ^i. For any two allocations μ^i, μ^j, we say that μ^i *dominates μ^j with respect to μ* if $S(\mu^j) \subseteq S(\mu^i)$ and for all $s \in \mathcal{S}$, $\mu^i(s) R_s \mu^j(s)$. In other words, one allocation dominates another with respect to the base allocation if all the students whose assignments change in one also have their assignments change in the other, and all students are at least as well off in the latter.

Selecting an Improvement

Let R be a preference profile and let μ be the allocation produced by the DA rule for some selection of courses C. We call μ the *base allocation at R for C*. Let $\mu^1, ...\mu^k$ be k the allocations resulting from k different improvements generated by the Improvement Step. For each μ^i, let $S(\mu^i) = \{s \in \mathcal{S} : \mu^i(s) \neq \mu(s)\}$ be the set of all students whose assignments change from μ to μ^i.

For any two μ^i, μ^j:

1. If μ^i dominates μ^j with respect to μ, we discard μ^j.

2. If not, we keep both courses.

[9] Or, alternatively, to select a set of courses that, when we run the DA procedure again on those courses, produces a constrained efficient allocation for that profile.

We repeat the above check for every remaining pair of improvements. We call the set that remains after elimination of dominated improvements the set of *valid improvements.* From this set, we pick an improvement arbitrarily.

Moreover, any selection from the set of valid improvements would get us to a constrained efficient allocation, at most by running the Assignment Step once more. To see this, note that if an allocation from the set of valid improvements were itself to have a improvement, then the latter improvement would also be a improvement over the base allocation. Thus it would have been generated in the Improvement Step, and the first improvement would have been discarded as a dominated improvement. Thus no allocation from the set of valid improvements is itself dominated, and so it is either constrained efficient, or running the Assignment Step for those courses produces a constrained efficient allocation.

7.2.1 Example

To continue with the example in the previous section, we have two improvements. The associated course selections are $C_1 = \{a, b, d\}$ and $C_2\{a, b, e\}$.

If we select the improvement C_1 and the courses $\{a, b, d\}$, we get the following allocation:

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	$\boxed{\text{d}}$	$\boxed{\text{a}}$	$\boxed{\text{d}}$	e	e	$\boxed{\text{b}}$
4	3	5	3	2	a	e	b	d	$\boxed{\text{b}}$	a
5	5	1	6	3	b	d	c	$\boxed{\text{a}}$	a	d
6	2	3	4	5	c	c	a	c	c	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

Instead, if we select $C_2 = \{a, b, e\}$, the allocation with courses $\{a, b, e\}$ is:

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	d	a	d	$\boxed{\text{e}}$	e	$\boxed{\text{b}}$
4	3	5	3	2	$\boxed{\text{a}}$	$\boxed{\text{e}}$	$\boxed{\text{b}}$	d	b	a
5	5	1	6	3	b	d	c	a	$\boxed{\text{a}}$	d
6	2	3	4	5	c	c	a	c	c	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

We see that $S(C_1) = \{1, 2, 3, 5\}$ and $S(C_2) = \{2, 4, 5\}$. Thus no improvement dominates the other, and so we can pick either selection of courses. However, we shall see in the next section that this would have strategic implications.

7.3 Properties of the DAI Rule

We see from the example that the resulting allocation is constrained efficient and fair. Theorem 2 tells us that this is always true.

Theorem **2.** *The DAI procedure is fair and constrained efficient.*

Proof: The properties of the DA rule guarantee fairness, and the absence of an improvement guarantees constrained efficiency by Theorem 1. ■

8 Strategic Aspects of the DAI Rule

In this section we explore the strategic properties of the DAI rule. Strategy-proofness is a property that ensures that it is a dominant strategy for all students to truthfully report their preferences. In a strategy-proof rule, no student can do strictly better by falsely reporting some other preferences, given everybody else's reported preferences. Formally:

Definition **8.** A mechanism f is *strategy-proof* if, for every profile of preferences R, every student s, and every report R'_s, we have that $f_s(R) R_s f_s(R'_s, R_{-s})$.

The DAI rule is not strategy-proof, even though the underlying DA rule is. Before we discuss where strategy-proofness fails, however, let us first see where it works. Theorem 3 shows that that no student can favourably induce the addition of a course or favourably change her assignment within the set of assigned courses via a manipulation.

Theorem **3.** *Let R be a profile and let $\mu = f^D(R)$ by the allocation produced by the DAI rule. Let $s \in \mathcal{S}$ be any student and $c \in \mathcal{C}$ a course such that $cP_s\mu(s)$. If $c \in C_\mu$ or if there is no improvement of μ that includes c, then $f^D(R'_s, R_{-s}) \neq c$ for all R'_s.*

The proof is in Appendix B. Essentially, we argue that if there is a student who prefers some course to her DAI assignment, she cannot get this course for any profile where she is the only student to report different preferences. We appeal to the fact that she is either 'too low' in the priority for that course to affect its assignment by ousting other students. Or, since she cannot affect other student preferences, she cannot unilaterally induce a improvement for an unassigned course that will grant it to her.

However, the rule is not entirely strategy-proof. There can be situations where a student can cause a potentially Pareto improving course to be removed from contention in favour of another. Note that she would not have any incentive to do this if she is not part of both improvements. But if she is part of both improvements, then she may be differently well off if one course is picked over the other. It is possible that an agent could render one of those improvements infeasible. If there were no other options, this would cause the rule to select the other course. And if this makes her better off, there is a potential for manipulation.

But as Theorem 3 demonstrates, this is the only way a student can manipulate. We show the manipulability of the rule by example.

8.1 Example

To continue the example in the previous section, let us reproduce in the tables below the allocations that would result if we selected courses $\{a, b, d\}$ or $\{a, b, e\}$ at the Improvement Step, and re-ran the Assignment Step:

If we select courses $\{a, b, d\}$, after the Assignment Step we would get the following allocation:

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	$\boxed{d}$	$\boxed{a}$	$\boxed{d}$	e	e	$\boxed{b}$
4	3	5	3	2	a	e	b	d	$\boxed{b}$	a
5	5	1	6	3	b	d	c	$\boxed{a}$	a	d
6	2	3	4	5	c	c	a	c	c	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

Instead, if we select $\{a, b, e\}$, the Assignment Step would produce:

$\succ_a$	$\succ_b$	$\succ_c$	$\succ_d$	$\succ_e$	P_1	P_2	P_3	P_4	P_5	P_6
1	6	2	1	4	d	$\boxed{\text{a}}$	d	$\boxed{\text{e}}$	$\boxed{\text{e}}$	$\boxed{\text{b}}$
4	3	5	3	2	$\boxed{\text{a}}$	e	$\boxed{\text{b}}$	d	b	a
5	5	1	6	3	b	d	c	a	a	d
6	2	3	4	5	c	c	a	c	c	e
2	1	4	5	6	e	b	e	b	d	c
3	4	6	2	1						

Note that agent 5's assignments are changing in both allocations. Thus he is part of an improvement in both cases. But agent 5 can manipulate. To see this, consider a preference R'_5 with cP'_5b. It is easy to see that the improvement with courses $\{a, b, d\}$ would no longer exist. Thus the only improvement would be with courses $\{a, b, e\}$, and the rule would select this one. But now agent 5 receives e which he prefers to b, which is what he was assigned in the other improvement. Thus agent 5 has an incentive to drop the improvement that gives him b in favour of the one that gives him e.

9 Conclusion

In this chapter we have formulated a fair and constrained efficient solution to the problem of allocating courses with exact capacity constraints to students based on preferences and priorities. In the process we have outlined a Pareto improvement procedure that allows us to find a constrained efficient allocation from any internally constrained efficient allocation, in a manner that preserves the fairness of the original allocation.

This is particularly useful in the context of the college in our motivating example. For any number of reasons, the college may have a pre-specified selection of courses that it would rather offer to the students. Using this as the default offering, our rule allows the college to replace some courses with others when necessary, especially when enough students would rather have some other course than one from their selection. The default selection of courses will be altered in a fair and efficiency-improving way.

The DAI rule is not strategy-proof. Yet the possibilities for manipulation are limited.

A future direction of research in this paper would be to consider the time complexity of the algorithm underlying the DAI rule. This would be useful especially when comparing the efficiency of the rule as compared to the brute force approach. We have already shown that running the procedure once is enough to guarantee a fair and constrained efficient allocation. The procedure involves two repetitions of the DA procedure, which is polynomial time. It is the building of the Pareto chains that will determine the time efficiency of the procedure. Erdil and Ergin (2008) show that their improvement algorithm is polynomial time, so a similar result in this case would be of some use as well.

We can extend the analysis in this chapter in multiple directions. It can be seen that the procedure we have outlined can also be used in cases where courses have different capacities from each other. Care must be taken however while constructing improvements in this case, as not every course can be replaced by any other course. Thus the set of feasible allocations will be altered in this scenario, and some courses may end up being complementary to others.

Another related extension would be to relax the requirements of exact constraints, and allow courses to have different minimum and maximum constraints. This would increase the set of feasible allocations and possibly allow the rule to gain even more in terms of efficiency.

References

Abdulkadiroğlu, A. and L. Ehlers (2006): "Controlled school choice," Mimeo.

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

Abizada, A. and S. Chen (2011): "The college admissions problem with entrance criterion," Mimeo, University of Rochester.

Alcalde, J. and S. Barberà (1994): "Top dominance and the possibility of strategy-proof stable solutions to matching problems," *Economic Theory*, 4, 417–435.

Dubins, L. and D. Freedman (1984): "Machiavelli and the Gale-Shapley algorithm," *American Mathematical Monthly*, 88, 485–494.

Ehlers, L., I. E. Hafalir, M. B. Yenmez, and M. A. Yildirim (2011): "School choice with controlled choice constraints: Hard bounds versus soft bounds," Mimeo.

Erdil, A. and H. Ergin (2008): "What's the matter with tie-breaking? Improving efficiency in school choice," *The American Economic Review*, 98, 669–689.

Ergin, H. I. (2002): "Efficient resource allocation on the basis of priorities," *Econometrica*, 70, 2489–2497.

Fragiadakis, D., A. Iwasaki, P. Troyan, S. Ueda, and M. Yokoo (2012): "Strategy-proof matching with minimum quotas," Mimeo.

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

Hatfield, J. W. (2009): "Strategy-proof, efficient, and non-bossy quota allocations," *Social Choice and Welfare*, 33, 505–515.

Kesten, O. (2006): "On two competing mechanisms for priority-based allocation problems," *Journal of Economic Theory*, 127, 155–171.

Kojima, F. and M. Manea (2010): "Axioms for Deferred Acceptance," *Econometrica*, 78, 633–653.

Pápai, S. (2000): "Strategyproof assignment by hierarchical exchange," *Econometrica*, 68, 1403–1433.

——— (2001): "Strategyproof and non-bossy multiple assignments," *Journal of Public Economic Theory*, 3, 257–271.

Pycia, M. and M. U. Ünver (2013): "Incentive-compatible allocation and exchange of discrete resources," Mimeo.

Roth, A. (1982a): "The economics of matching: stability and incentives," *Mathematics of Operations Research*, 7, 617–628.

——— (1982b): "Incentive compatibility in a market with indivisible goods," *Economics Letters*, 9, 127–132.

Shapley, L. and H. Scarf (1974): "On cores and indivisibility," *Journal of Mathematical Economics*, 1, 23–37.

Sönmez, T. and T. Switzer (2011): "Matching with branch-of-choice contracts at United States Military Academy," Mimeo, Boston University.

Svensson, L.-G. (1999): "Strategy-proof allocation of indivisible goods," *Social Choice and Welfare*, 16, 557–567.

10 Appendix A: Proof of Theorem 1

Let R be a preference profile, let μ be a feasible, fair and internally constrained-efficient allocation at a selection of courses C_μ.

10.0.1 Sufficiency

In one direction, suppose there exists a selection of courses C such that, for $C' = C \setminus C_\mu$, there is an improvement $\Gamma^{C'}$ such that $\bar{\gamma}^s \in C_\mu \setminus C$ for all $\gamma \in \Gamma^{C'}$. Let μ' be the upgrade of μ via $\Gamma^{C'}$. Since $\bar{\gamma}^s \in C_\mu \setminus C$ for all $\gamma \in \Gamma^{C'}$, and $\bar{\gamma}_i \neq \bar{\gamma}_j$ for all $i \neq j$, we have that if $c \in C_\mu \setminus C$, then $c \notin C_{\mu'}$. Thus $C_{\mu'} = C$, and μ' is feasible. By construction, μ' is fair, and Pareto dominates μ. Thus μ is not constrained efficient.

Necessity

We first prove a useful lemma.

Lemma **1.** *Let μ and ψ be fair assignments, let ψ be constrained efficient, and let ψ Pareto dominate μ, i.e., $\psi(s)R_s\mu(s)$ for all s, and $\psi(t)P_t\mu(t)$ for some t. Then:*

(a) For every course $c \in C_\psi$, we have that $|D_\mu(c)| \geq q$.

(b) Let $c \in C_\psi$, let $s = top(\succ_c, D_\mu(c))$, and suppose that $c \neq \mu(s)$. Then $\psi(s)R_sc$, and in particular, $\psi(s) \neq \mu(s)$.

Proof: Let μ and ψ be fair assignments, let ψ be constrained efficient, and let ψ Pareto dominate μ, i.e., $\psi(s)R_s\mu(s)$ for all s, and $\psi(t)P_t\mu(t)$ for some t.

(a) Pick some $c \in C_\psi$, and let $S_c = \{s \in \mathcal{S} | \psi(s) = c\}$. We have that $\psi(s)R_s\mu(s)$ for all $s \in S_c$ since ψ Pareto dominates μ. Thus $s \in D_\mu(c)$ for all $s \in S_c$, and since $|S_c| = q$, we have that $|D_\mu(c)| \geq q$.

(b) Suppose $cP_s\psi(s)$. Since $s = top(\succ_c, D_\mu(c))$, we have in particular that $s \succ_c s'$ for all $s' \in \psi(c)$. This violates fairness of ψ. Thus $\psi(s)R_sc$. Since ψ Pareto dominates μ, and $c \neq \mu(s)$, it follows that $\psi(s) \neq \mu(s)$.

■

Suppose μ is not constrained efficient. Then there exists a constrained efficient allocation ψ that Pareto dominates μ. Let C be the courses at ψ, and let $C' = C \setminus C_\mu$. Note that we have $C_\psi \neq C_\mu$, because otherwise ψ is a rearrangement of μ, violating the internal constrained efficiency of μ. We will construct $\Gamma^{C'}$.

Initialisation:

1. Let ν be a tracking allocation and set $\nu = \mu$. Let D_ν be the associated claimant profile.
2. Let $C' = \{c_1, ..., c_r\}$, and for every $c_i \in C'$, let $\{o_1^{c_i}, ..., o_q^{c_i}\}$ be the slots in c_i.
3. For each course $c \in C'$ and a slot o_j^c, create a chain γ by setting the opening node $\gamma^c(0) = c$.
4. Set $i = 1, j = 1, k = 1$.

Construction Step:
Round k:

1. Let $s_{ij}(k) = top(\succ_{\gamma_{ij}^c(k-1)}, D_\nu(\gamma_{ij}^c(k-1)))$.

   Claim **1.** For any $k \geq 1$, $D_\nu(\gamma_{ij}^c(k-1)) \neq \phi$.

   Proof: In any round, ψ Pareto dominates ν (see Claim 2 below). Thus for every round k, $|D_\nu(\gamma_{ij}^c(k-1)| \geq q$, by Lemma 1(a). ■

2. Update $\nu(s_{ij}(k)) = \gamma^c_{ij}(k-1)$

   Claim **2.** The temporary allocation ν is fair. Also, ψ Pareto dominates ν.

   Proof: In the first round, μ is fair so ν is fair. In every subsequent round, ν is fair because if a student's assignment changes from the previous round, she must be the top claimant for the course she is newly assigned, and so no other student envies her. In the first round, ψ Pareto dominates μ. In any subsequent round k, since by Lemma 1(b) we have that $\psi(s) R_s \nu(s)$ for all students in round $k-1$, the same is true at round k also after the transfer. We show that the relation is strict for some student. Since $\gamma^c_{ij}(k-1) \notin C_\mu \setminus C$ in a non-terminal round, there must be a student s such that $\nu(s) \in C_\mu \setminus C$. By construction, $\nu(s) = \mu(s)$. Thus by Lemma 1(b), we have that $\psi(s) P_s \nu(s)$ in the previous round, and thus in this round too since her assignment does not change. So ψ Pareto dominates ν.

   ■

3. Set $\gamma^s_{ij}(k) = s_{ij}(k)$ and $\gamma^c_{ij}(k) = \mu(s_{ij}(k))$.

   Claim **3.** $s_{ij}(k) \neq s_{ij}(l)$ for all $l < k$.

   Proof: Suppose not. Consider $s_{ij}(l), s_{ij}(l+1), ..., s_{ij}(k) \equiv s_{ij}(l)$. By construction, each student prefers the earlier student's assignment, which is in μ. Thus we have a Pareto cycle in μ, and μ is not internally constrained efficient, which is a contradiction. ■

4. If $\gamma^s_{ij}(k) = \gamma^s_{i'j'}(k')$ and $\gamma^c_{ij}(k) = \gamma^c_{i'j'}(k')$ for some i', j', k', then remove $\gamma^s_{i'j'}(k'')$ and $\gamma^c_{i'j'}(k'')$ for all $k'' \geq k'$.

   Claim **4.** If $\gamma^s_{ij}(k) = \gamma^s_{i'j'}(k')$ and $\gamma^c_{ij}(k) = \gamma^c_{i'j'}(k')$, then $\gamma^s_{ij}(k+m) = \gamma^s_{i'j'}(k'+m)$ and $\gamma^c_{ij}(k+m) = \gamma^c_{i'j'}(k'+m)$ for all $m \geq 0$.

   Proof: Suppose $\gamma^s_{ij}(k) = \gamma^s_{i'j'}(k')$ and $\gamma^c_{ij}(k) = \gamma^c_{i'j'}(k')$. Then we have that $top(\succ_{\gamma^c_{ij}(k+1)}, D_\nu(\gamma^c_{ij}(k+1))) = top(\succ_{\gamma^c_{i'j'}(k'+1)}, D_\nu(\gamma^c_{ij}(k'+1)))$. Thus $\gamma^s_{ij}(k+1) = \gamma^s_{i'j'}(k'+1)$ and $\gamma^c_{ij}(k+1) = \gamma^c_{i'j'}(k'+1)$. Repeating for all $m \geq 1$, we have the desired result. We remove all duplicated students from $\gamma_{i'j'}$ because by construction, $\gamma^c_{ij}(k) P_{s_{ij}(k)} \gamma^c_{i'j'}(k')$, and so she must get at least $\gamma^c_{ij}(k)$ by Lemma 1(b). ■

If $\gamma^c_{ij}(k) \in C \cap C_\mu$, we go to Round $k+1$. Otherwise $\gamma^c_{ij}(k) \in C_\mu \setminus C$, and we go to the Verification Step.

<u>Verification Step:</u>

1. We repeat for all slots in a course, i.e., if $j < q$, then we increment j, set $k = 1$, and go back to the Construction Step.
2. We repeat for all courses, i.e., if $j = q$ but $i < r$, we increment i, set $j = 1, k = 1$, and go back to the Construction Step.
3. If there is a course c_i and a slot $o^{c_i}_j$ such that, in the relevant chain γ_{ij}, we have that $\bar{\gamma}_{ij} \notin C_\mu \setminus C$, we repeat the Construction Step for that course.

Claim **5.** The process is guaranteed to terminate in a finite number of steps.

Proof: The number of courses and students is finite, and at every step some student is strictly improving her assignment. ■

When the process completes, we have the improvement $\Gamma^{C'}$ as required, with each chain originating in a new slot, and terminating in a discarded course.

11 Appendix B: Proof of Theorem 3

Lemma **2.** *Let R be a profile, let $\mu = f^D(R)$, and consider a student $s \in \mathcal{S}$ and some course $c \in \mathcal{C}$ such that $cP_s\mu(s)$. For some R'_s, let $\mu' = f^D(R'_s, R_{-s})$. If $C_{\mu'} = C_\mu$, then $\mu'(s) \neq c$.*

Proof: Since $C_{\mu'} = C_\mu$, μ' is produced by the rule for the same selection of courses as in μ. The result follows from the strategy-proofness of the DA rule for any fixed selection of courses. ■

Consider an allocation $\mu = f(R)$, a student $s \in \mathcal{S}$, and some course $c \in \mathcal{C}$ such that $cP_s\mu(s)$. Consider any R'_s, and let $\mu' = f^D(R'_s, R_{-s})$. We will show that $\mu'(s) \neq c$. There are two possibilities: Either $c \in C_\mu$ or not. We examine each in turn.

Suppose $c \in C_\mu$. Then $s \notin top_q(\succ_c, D_\mu(c))$. By Lemma 2, we have that $\mu'(s) \neq c$ if $C_{\mu'} = C_\mu$. So let $c' \in C_{\mu'} \setminus C_\mu$. Since c' is not assigned in μ, and only student s changes her preferences, it must be that (1) $|D_\mu(c')| < q$, and (2) $s \in D_{\mu'}(c')$. Thus $\mu'(s)R_s c'$ by Lemma 1(b). If $\mu'(s) \neq c'$, then by Lemma 1(a) we have that $|D_\mu(c') \setminus \{s\}| \geq q$, which is a contradiction. Thus $\mu'(s) = c'$ and, in particular, $\mu'(s) \neq c$.

Now, suppose instead that $c \notin C_\mu$. Then we have that $s \in D_\mu(c)$. But c is not assigned in μ, and there is no improvement of μ that includes c, so there must be at least one Pareto chain γ associated with some slot $o^c \in c$ such that $\bar{\gamma}^c \in C_\mu$ and $|D_\mu(\bar{\gamma}^c)| = q$. Thus there is no extra claimant for $\bar{\gamma}^c$, which student s cannot affect.

Since R'_s was arbitrary, s cannot get c for any R'_s.