# DEFERRED ACCEPTANCE

The point of this note is to take you through the logic behind the deferred acceptance algorithm. Please read the papers associated with the links on the web page for motivation and applications.

**Matching.** This algorithm is used to handle a number of different *matching problems* in which individuals are matched with organizations. For what we'll discuss here, the matching problem has to satisfy two properties: first, though the matching may involve money transfers, the transfers must be fixed at the time the matching takes place; second, each side of the market must have a finite capacity for the number of partners. For example admitting students to schools fits into this framework because each student only wants to get into one school, and schools can only admit as many students as they have seats in their classrooms.

In one sense every trading problem can be viewed as a matching problem where buyers are matched with sellers. However, in many of the trading problems you are familiar with, like auctions, money transfers are altered during the trading process to try to alter the matching (for example, a bid in an auction). Furthermore, 'sides' of the market often don't have real capacity constraints. For example, a platform like apple or android tries to match people with apps like Facebook or Spotify. However, Spotify has no effective limit on the number of users it can match with. In these two example, deferred acceptance has nothing to contribute.

A matching problem that does fit the algorithm is admitting graduating students into post-graduate programs. We'll use this problem as an illustration in this reading. This is a many to one matching problem in the sense that each 'applicant' is admitted to only one 'organization', but each organization will admit many applicants.

In this note, we'll restrict to many to one matchings, and in the examples, one to one matchings, to make the notation a bit easier.

We'll use the notation $I$ to refer both to the set of applicants and the number of applicants. We'll speak about applicant $i$ to refer to a specific applicant. The set of organizations will be referred to as $J$, with a specific organization referred to as organization $j$.

Each of the organizations has a number of vacant positions $q_j$. It might not want to fill all of them if it doesn't like all the applicants, but it can never fill more than $q_j$ positions. A *matching* $\mu$ is a correspondence that takes each element of $I \cup J$ and names the set of individuals or organizations that element is matched with. For example, $\mu(i) \in J \cup \emptyset$ for each applicant $i$. If $\mu(i) = \emptyset$, that means that applicant $i$ remains unmatched.

Similarly $\mu(j) \subset I \cup \emptyset$. A feasible matching is one for which $|\mu(j)| \leq q_j$ for every organization $j$. (The notation $|A|$ means the number of elements in the set $A$).

We'll assume that each $i \in I$ can strictly rank each of the organizations in $J$ in the sense that a higher ranked organization is one that the applicant would prefer to a lower ranked organization - prefer means 'would rather be accepted by'. We'll assume the ranking is strict. If $j$ is ranked higher (preferred) by $i$ to $j'$, we'll write

$j \succ_i j'$. We also allow that an applicant might prefer to remain unmatched than to be assigned to some organization, in which case we write $\emptyset \succ_i j$

We'll do the same for organizations and assume they strictly rank applicants with $i \succ_j i'$ meaning that organization $j$ would prefer to accept $i$ than to accept $i'$ if it can't accept them both. An organization $j$ might be unwilling to accept an applicant $i$, in which case we write $\emptyset \succ_j i$. This way of thinking about organization preference is actually pretty special, since organizations preferences over applicants will typically depend on who is already in the organization. In most of the literature preferences like this are called *regular*.

Once a matching has been made, we want to make sure that people won't want to go around changing the outcome. Of course, everyone wants to get into the best school, which has a limited capacity to accept students. So no matching will every give everyone their favorite outcome. Instead we ask for a somewhat weaker property - that a matching be *stable*.

A matching is stable if, whenever an applicant $i$ prefers some alternative $j'$ to the organization $\mu(i)$ to which they were assigned, then either that organization is unwilling to accept $i$ in the first place, or that organization has been assigned a group of applicants who fill its capacity, all of whom are preferred to $i$. Similarly, if an organization $j$ prefers some applicant $i'$ to an applicant $i \in \mu(j)$, then applicant $i'$ prefers $\mu(i')$ to $j$. If a matching is stable, everyone might wish they were matched with someone else, but they won't bother trying to change anything because they know they won't be able to get anyone they to match with them.

Just to get a sense of what stability means, suppose there are only two applicants $a$ and $b$ and two organizations, say UBC and SFU where they could be accepted. In this example, UBC and SFU both have one seat to fill. Applicant $a$ prefers UBC to SFU while for applicant $b$ it is the other way around. UBC prefers applicant $b$ to applicant $a$, while SFU prefers $a$ to $b$. There are two possible matchings in this example - $a$ is accepted by UBC while $b$ is accepted at SFU, or $a$ is accepted by SFU while $b$ is accepted by UBC.

In the first matching UBC would prefer to have applicant $b$, but $b$ prefers SFU, who she gets in the matching, to UBC, so she won't leave. In the second matching, $a$ would prefer to be at UBC, but UBC is already assigned to $b$ which it prefers.

**Deferred Acceptance.** Notice that is this example illustrates that there will sometimes by many stable matchings. However, at this point, all we want to do is to find one of them. To do it, we can use the deferred acceptance algorithm. We'll describe the applicant proposing version.

For applicants, behavior is simple - applicants can apply only to schools who have not rejected them. That seems straightforward enough. To make it into an algorithm, we'll have a computer program that asks the applicants for their preferences over institutions, then applies for them. Each time it applies, it will send the application to the organization which the applicant said was most preferred among the set of organizations who have not yet rejected her application.

For organizations, we'll do something similar. We'll ask them to describe their preferences over applicants, then whenever an application comes in, we'll have the computer program reject it if the organization said that the applicant was unacceptable under any circumstances, otherwise, the program will hold the application until a better one comes along. What it means for a better one to come along is that the organization's capacity is filled by the applications it is holding and the new applicant is better than one of the applications it is currently holding.

To see it in action, you need to start with the preferences of the organizations and the applicants. Lets do an example with three organizations, UBC, SFU and UT and four students $a$, $b$, $c$, and $d$. SFU claims that its preferences are

$$SFU : a \succ b \succ c \succ d$$

while for UBC:

$$UBC : b \succ a \succ c \succ d$$

and finally

$$UT : c \succ b \succ d \succ a$$

For the students we elicit

$$\begin{array}{rl} a: & UBC \succ UT \succ SFU \\ b: & UBC \succ SFU \succ UT \\ c: & UBC \succ SFU \succ UT \\ d: & UBC \succ UT \succ SFU \end{array}$$

We'll also assume that each applicant would prefer to be in any of the three institutions rather than not going to school at all, while each university is willing to accept any of the students if it has space.

We can make the exercise a bit easier to follow by assembling the information above into a little table that looks like the table we used to describe Nash equilibrium. It is important to remember that isn't what this table does. It just mimics the rankings given above for the applicants and organizations. The rankings of the organization over the applicants are given as the first entry in each cell, the second entry is the corresponding ranking of organizations by the applicants. The lower the number the more highly the applicant or school is ranked

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3 | 2,2 | 3,2 | 4,3 |
| UBC | 2,1 | 1,1 | 3,1 | 4,1 |
| UT | 4,2 | 2,3 | 1,3 | 3.2 |

Now the applicants start to apply

- At each iteration of this algorithm, each applicant sends an application to their most preferred school in the set of schools that have not yet rejected them. In our example, all the students like UBC the best so all the applications go there. The applications are marked with asterisks in the next table:

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3 | 2,2 | 3,2 | 4,3 |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2 | 2,3 | 1,3 | 3,2 |

In this first iteration, UBC receives four applications, and it likes the one from $b$ the best, so it rejects the other three. Lets mark the rejected applications in red. The rejected applicants $a$, $c$, and $d$ cannot make an application to UBC again.

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3 | 2,2 | 3,2 | 4,3 |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2 | 2,3 | 1,3 | 3,2 |

We don't yet have a full matching, so we do a second iteration. After $a$'s rejection by UBC, $a$'s favorite school is UT, so he should apply there. Similarly $c$ has been rejected by UBC and prefers SFU. Then $d$ prefers UT, so we just mark the new applications with asterisks.

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3 | 2,2 | 3,2* | 4,3 |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2* | 2,3 | 1,3 | 3,2* |

To complete this iteration, UT has two applications and it prefers the application from $d$ to the one from $a$, so it rejects $a$. SFU continues to hold the application from $c$, which completes the iteration

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3 | 2,2 | 3,2* | 4,3 |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2* | 2,3 | 1,3 | 3,2* |

On the third iteration, $a$ is the only applicant who does not have an accepted application. He has only one option, apply to SFU, which is his least favorite school. At this point SFU has two applications one from $a$ and one from $c$ that it has held since the last iteration. Since it prefers $a$, it rejects the application from $c$.

This is why the algorithm is called 'deferred acceptance' - SFU accepted $c$ initially, but only until a better alternative came along.

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3* | 2,2 | 3,2* | 4,3 |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2* | 2,3 | 1,3 | 3,2* |

On the final iteration, $c$ applies to UT. Since $c$ is preferred to the incumbent $d$, $d's$ application is rejected, $c's$ is accepted and $d$ is just out of luck since she doesn't have an option that hasn't rejected her.

| | a | b | c | d |
|---|---|---|---|---|
| SFU | 1,3* | 2,2 | 3,2* | 4,3* |
| UBC | 2,1* | 1,1* | 3,1* | 4,1* |
| UT | 4,2* | 2,3 | 1,3* | 3,2* |

Don't think of this process as actual applications. This is all done in the blink of an eye by a computer program.

**Algorithms.** The point of all this want to try to compute a matching that was stable. It seems to work pretty well because you'll notice that at each iteration, at least one application is rejected. Since there is a finite number of students and schools, this guarantees that the algorithm will stop in finite time.

The natural question to ask is what do you get when it stops?

The answer is that the matching that results when it stops must be a stable matching. To show this we use a proof by contradiction.

Suppose that once the algorithm stops, we can find a blocking pair, an organization $j$ and an applicant $i$ for whom $i \succ_j \mu(j)$ while $j \succ_i \mu(i)$. Notice that during the deferred acceptance algorithm, applicant $i$ always has to apply to his or her favorite organization among all the ones that haven't rejected them. If that is true and $i$ is matched with $\mu(i)$ which she doesn't like as much, then that could only mean that $i$ applied to $j$ and was rejected. Yet if $j$ rejected $i$, it could only be because it already had an application from an applicant it preferred. Since it would subsequently only accept applicants that were even better, it has to be that $\mu(j) \succ i$, which is a contradiction.

**Implementing the Deferred Acceptance Solution.** The market with which I am most familiar is the market for junior academic economists. The applicants are economics students graduating with Ph'd degrees. The organizations are Universities, Governments, and Consulting firms throughout the world. The market runs through much of the year, but the primary hiring period begins in October when Universities place job ads. In 2014 there were a little more than 4000 applicants registered at econjobmarket.org, one of the main advertising and recruiting sites. The number of academic openings is somewhere between 1200 and 1600. Junior economists apply to schools they are interested in. The schools review applications, interview applicants at a centralized meeting in January, then make offers to the applicants they like. The applicants, many of whom receive multiple offers, choose the offer that they like.

This might give you some idea about the potential size of a market, and how complex it might be to find some kind of stable matching. It mostly illustrates that it won't be feasible to find a good solution by scribbling on pieces of paper, as we did above.

In 2013, the Vancouver School Board implemented a matching program for children in special kindergarten programs. There were four programs, French Immersion, Mandarin Immersion, Montessori, and Fine Arts, spread over 19 schools. In the year prior to their program, about 1400 students participated. The school board discussed the possibility of matching students using deferred acceptance, but later changed their mind and used a simple computerized lottery using only parents' first choice. This system replaced one in which parents physically traveled to as many schools as they wanted and submitted an application. Schools then ran their own lotteries.

Neither of these examples actually use the deferred acceptance algorithm. In the school board case, the board was concerned about the difficulty parents might have expressing their preferences over schools as well as with the fact that parents have a strong tendency to change their minds and give up their matching once they get it. In the the academic job market, universities don't have clear preferences at the start of the process and need time to develop them. There is no hope of implementing a pareto optimal and stable matching if you don't know participants preferences.

The Vancouver school board example is an interesting case in point. The way their current matching program works is that each parent lists three schools in order or their preference. The actual lottery only makes use of the first choice. Each child is entered into a draw for their parents first choice program. If they win the draw, that is great. If they don't, they are put on a waiting list for that school in case another family who did win the lottery decides to decline. Since parents who listed a school as their first choice are always put on a waiting list, the second and third choice are basically not attainable unless these choices end up with unfilled spots. If parents are very anxious to get their children into a special program, they need to choose the program where they think they are most likely to get in. This may or may not be the one they most prefer for their child.

This is one of the reasons that deferred acceptance seems advantageous. The side who make proposals (parents in the case of the Vancouver School Board, students in the college application example) have no reason to mis-represent their preferences the way they do with the Vancouver School Board example. That would be great except that, as I mentioned, parents don't always know their preferences.

It isn't always in everyone's interest to submit their preferences truthfully. For example, suppose we look at a simpler problem with only SFU and UBC and only two students, 1 and 2. We'll add the option of not going to university and not filling an open slot. Suppose that preferences look like this

$$UBC \to 1 \succ 2 \succ OUT$$

$$SFU \to 2 \succ 1 \succ OUT$$

while the students' preferences are

$$1 \to SFU \succ UBC \succ OUT$$

and

$$2 \to UBC \succ SFU \succ OUT$$

The student proposing algorithm has 1 applying to SFU and 2 applying to UBC. That is it, there are no more proposals. So $\mu(1) = SFU$, $\mu(2) = UBC$. The school proposing algorithm has UBC proposing to 1 and SFU proposing to 2, so the matching is $\mu(1) = UBC$, $\mu(2) = SFU$. Notice that both the schools are better off in the school proposing algorithm than they are in the student proposing algorithm. I described this example earlier.

Now notice that something strange happens in this example. Suppose that in the student proposing version, that UBC simply lies and says that they are unwilling to accept 2 under any circumstances. In other words, it submits the preference $UBC \to 1 \succ OUT$. Lets run the student proposing version of the algorithm - 2 proposes to UBC and 1 proposes to SFU. At this point, our computerized algorithm decides that UBC should reject 2's application. Now 2 is forced to apply at SFU where she is preferred, causing 1's application to SFU to be rejected. Then 1 applies at UBC and the match is complete. Notice that by misrepresenting its preferences, UBC ends up with a student that they prefer (as does SFU).

0.1. **A slightly more complicated problem:** Lets make things slightly more complicated by adding a few more students, another university, and another option, The additional option will be not going to university at all. Here are the preferences. At this point you can imagine that these are preferences that were actually submitted through the web page. The problem is just to do the matching. Here are the preferences written in two different ways:

$$SFU \to 1 \succ 4 \succ 2 \succ 3$$

$$UBC \to 1 \succ 3 \succ 4 \succ 2$$

$$UOT \to 2 \succ 3 \succ 1 \succ 4$$

$$UFV \to 2 \succ 3 \succ 4 \succ 1$$

$$1 \to UBC \succ SFU \succ UOT \succ UFV \succ OUT$$
$$2 \to UBC \succ UOT \succ SFU \succ UFV \succ OUT$$
$$3 \to UBC \succ SFU \succ UOT \succ UFV \succ OUT$$

$$4 \to UBC \succ UOT \succ SFU \succ OUT \succ UFV$$

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1 | 4,1 | 2,1 | 3,1 |
| SFU | 1,2 | 3,3 | 4,2 | 2,3 |
| UFV | 4,4 | 1,4 | 2,4 | 3,5 |
| UOT | 3,3 | 1,2 | 2,3 | 4,2 |
| OUT | -,5 | -,5 | -,5 | -,4 |

We've added a fourth university (if you like name UFV means University of the Fraser Valley). There is also a fourth student who only wants to go to UBC, SFU or UOT, but is not interested in UFV.

Lets find a stable matching using the algorithm:

(1) All students propose to UBC, which tentatively accepts student 1 and rejects the other proposals;
(2) Students 2 and 4 propose to UOT, while student 3 proposes to SFU, SFU tentatively accepts student 3, UOT tentatively accepts student 2 who it prefers;
(3) Student 4, who has been rejected by UBC and UOT, now proposes to his remaining favorite SFU. SFU prefers 4 to their current acceptance, 3, so they throw 3 out and accept 4;
(4) Student 3 has been rejected by UBC and now SFU, so he proposed to his next favorite, UOT who has currently accepted 2 who they prefer, student 3's application is rejected;
(5) Finally, student 3 proposes to UFV who accepts him.

This gives us

$$\mu(1) = UBC, \mu(2) = UOT, \mu(3) = UFV, \mu(4) = SFU.$$

Now in pictures:

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1* | 4,1* | 2,1* | 3,1* |
| SFU | 1,2 | 3,3 | 4,2 | 2,3 |
| UFV | 1,4 | 1,4 | 2,4 | 3,5 |
| UOT | 3,3 | 1,2 | 2,3 | 4,2 |
| OUT | -,5 | -,5 | -,5 | -,4 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1* | 4,1* | 2,1* | 3,1* |
| SFU | 1,2 | 3,3 | 4,2* | 2,3 |
| UFV | 1,4 | 1,4 | 2,4 | 3,5 |
| UOT | 3,3 | 1,2* | 2,3 | 4,2* |
| OUT | -,5 | -,5 | -,5 | -,4 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1* | 4,1* | 2,1* | 3,1* |
| SFU | 1,2 | 3,3 | 4,2* | 2,3* |
| UFV | 1,4 | 1,4 | 2,4 | 3,5 |
| UOT | 3,3 | 1,2* | 2,3 | 4,2* |
| OUT | -,5 | -,5 | -,5 | -,4 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1* | 4,1* | 2,1* | 3,1* |
| SFU | 1,2 | 3,3 | 4,2* | 2,3* |
| UFV | 1,4 | 1,4 | 2,4 | 3,5 |
| UOT | 3,3 | 1,2* | 2,3* | 4,2* |
| OUT | -,5 | -,5 | -,5 | -,4 |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| UBC | 1,1* | 4,1* | 2,1* | 3,1* |
| SFU | 1,2 | 3,3 | 4,2* | 2,3* |
| UFV | 1,4 | 1,4 | 2,4* | 3,5 |
| UOT | 3,3 | 1,2* | 2,3* | 4,2* |
| OUT | -,5 | -,5 | -,5 | -,4 |

**Exercise:**

(1) Find the stable outcome associated with the university proposing deferred acceptance algorithm for this last example. Are they the same? Can you see why this means that the stable matching is unique in this case? Can you see why neither schools or students want to misrepresent their preferences in this case?

(2) Suppose all the students agree on the ranking of the schools (pick your favorite ranking). What happens? Is there a difference between the student proposing and the school proposing outcome?

(3) As special case of the above occurs when all the schools agree on the ranking of students, and all the students agree on a ranking of schools. What happens then? This outcome is called 'assortative matching'.

**Indifference.** Here is a problem that looks much like the ones we have done above:

| | 1 | 2 | 3 |
|---|---|---|---|
| UBC | 3,1 | 1,2 | 2,1 |
| SFU | 2,2 | 3,1 | 1,3 |
| UT | 3,3 | 2,3 | 1,2 |

The main difference here is that the 'preferences' of the universities are generated by school specific lotteries. That means that each school independently runs a draw that assigns values to each of the students. Student 1 does badly at each of the schools, placing no higher than 2 at each of them. Student 3 wins outright at both SFU and UT, but only comes second at UBC, which is her favorite. Student 2 wins the draw at UBC.

Now what happens? 1 and 3 apply first at UBC, where 1 is rejected. 2 applies first at SFU where she is tentatively accepted. Since 1 is rejected at UBC, he applies at SFU where he displaces 2 who is rejected. 2 then applies at UBC where she displaces 3, who ends up at UT.

This is actually quite a bad outcome. All three students ends up with their second choices, which doesn't sound good. What makes this even worse is that the only reason that they do is because the schools ran independent lotteries. In these lotteries, UBC ended up favoring 2 and SFU favored 1. Since UBC and SFU don't really care which students they get, we could make everyone better off by just switching schools for 1 and 2. As we would say, the matching is not even *pareto optimal*.

Oddly enough, the problem with this matching is entirely the fault of the independent lotteries. If instead you are to use a system wide ranking of students by schools (all schools rank the students the same way) then the resulting outcome would have to be pareto optimal. In the example above, student 1 would rather go to UBC than to SFU but UBC rejects him because he did poorly in their lottery.

On the other hand, SFU, which student 1 doesn't really want, rated student 1 very highly in their lottery. It is UBC that *blocks* the switch because it appears UBC is made worse off by accepting 1 instead of 2. Yet UBC isn't really made worse off by the switch, the preference is just the result of an arbitrary lottery. With a system wide lottery, if SFU prefers 1 over 2, then UBC must also prefer 1 over 2, and would be willing to accept them.

As an exercise, try doing the deferred acceptance algorithm with all three common rankings of the applicants, and verify that the outcome is pareto optimal in each case. The downside of the common system wide lottery is that when a student is given a low rank by one school, she is given a low rank by all schools.

**Vancouver School Board - Changes in Preferences.** The url jupyter notebook provides an illustration of the Vancouver School Board Kindergarten Special Programs matching. There are 18 schools with a total 600 seats in the simulation described in that jupyter notebook. There were 933 applicants to the special programs draw. The notebook illustrates how the deferred acceptance algorithm can be run on some simulated preference data from 2019 to generate a stable matching.

One of the problems that creates considerable work for them is that parents' preferences appear to change over time. In particular many parents either opt for private schools after the draw, or leave the country. The draw they use allocates seats to these parents who subsequently refuse them. The schools board is then forced to try to reallocate the seats after the initial draw.

The method they use for doing this is not deferred acceptance. Yet before I get back and try to describe it, I'll illustrate how the algorithm can be applied to problems like this. Deferred acceptance is a 'static' mechanism - it allocates students to spots, then doesn't do anything else.

For example, if we go back to the example we studied above

| | 1 | 2 | 3 |
|---|---|---|---|
| UBC | 3,1 | 1,2 | 2,1 |
| SFU | 2,2 | 3,1 | 1,3 |
| UT | 3,3 | 2,3 | 1,2 |

This results in each student getting their second favorite school choice as we discussed previously.

Now suppose that 1 changes his mind and decides to take a job instead. Since 1 is allocated to SFU, a spot there becomes vacant, and 2 would like to fill it. That seems simple enough, but if 2 moves to her favorite school SFU, then UBC becomes vacant and 3 would like to take that spot.

As you see, a little bit of shuffling can trigger side effects that trigger a lot more shuffling. Of course, it is possible at that point just to start the whole process over again without student 1 and re-run the deferred acceptance algorithm. Since withdrawals by students don't happen immediately, this seems disruptive. Any student who isn't placed in their favorite school can't be entirely sure of their placement until some time elapses.

This is one reason to implement the algorithm in stages using 'exploding' offers. An exploding offer is simply an offer with an expiration date - if it isn't accepted by that date, it is treated as a refusal.

The first step is just to run the deferred acceptance algorithm as if it were static. The only preference changes we anticipate are ones in which students who have applied to the program refuse their offers, so any student who is placed in a school on this initial pass is guaranteed to get a place among their three choices.

However, except for those who are placed in their first choice schools right away, the actual placement of many students will depend on who drops out. Furthermore, some of the students who aren't matched may also end up with offers if some of the existing students drop out.

In week 1, offers are made to all the students who have been placed in their first choice schools. All others who are matched by the deferred acceptance algorithm, but aren't given their first choice are simply told that they will receive an offer from one of their three choices within, say, three weeks. Any student who is unmatched is put on a system wide wait list.

At the end of this first week, a number of offers will have been rejected, opening slots that would otherwise have been taken. In addition, many schools won't have filled their capacity because initial offers are only made to students who are placed in their first choice school.

It is then straightforward to determined the next set of offers using the deferred acceptance procedure. The algorithm is run again using only the empty school slots, and the students who haven't yet been matched. At this point exploding offers should be made to students who are placed in either their first or second choice schools, again providing them with a chance to drop out before further offers are made.

At the end of the second week, a number of offers will have been rejected, leading to the final allocation being made in the third week when the deferred acceptance algorithm is run using the remaining empty slots and unplaced students.

**An algorithm that runs continuously.** Beyond week 3, there is no reason to stop this automated reallocation. Nor is there any real reason to have three initial stages, reallocations can be made on the fly as long as students continue to refuse their offers. This is how it works:

(1) Using initial preferences, take every student both matched an unmatched and associate with that student the school that they would most prefer to their existing match, call it $p(i)$ for notational purposes. Presumably for students who have already been matched to their most preferred school $p(i) = \emptyset$.
(2) If any student withdraws from school $\dot{j}$, then find all of the students $i$ such that $p(i) = j$, and make an exploding offer to whichever student in this

group has the highest rating at school $j$. If $i$ rejects this offer, change $p(i)$ to be $\emptyset$. Then repeat the process with the next highest rated student until it is accepted or has been rejected by all the students.

(3) If, in step 2, an offer is accepted, resulting in an opening in another school, say $k$, then go back to step 1 and repeat the process for school $k$.

What this algorithm does is to ensure that whenever the system is still (no offers are being made), the matching will be the student proposing deferred acceptance outcome for the remaining set of schools and students.

**What the Vancouver School Board does.** The Vancouver School Board uses a method that resembles the one above in the sense that each family is allowed to specify 3 schools in order of their preference. The school board has some preferences over students. It wants to give preference to children who have older siblings in the school, for example. Some of the schools impose what they call 'catchment' constraints in which they give preference to student who live in the neighborhood of the school (which are called catchments).

After grading students this way, the the school board takes the students who select each school as one of their choices, then sorts them into groups, those who selected the school as their first choice, those who selected the school as their second choice, and those who chose the school as their third choice. Each school then runs a lottery over the students in each group and assigns students who made the school their first choice to seats in the school in order of their rank until the seats are all filled.

If their aren't enough first choice students to fill all the seats, the school then goes to the second choice students and assigns them to seats until all the seats are filled. If there aren't enough first and second choice students to fill all the seats, the schools goes to its third choice students.

This method has a number of problematic features.

The first one is that students who named the school as their first choice are ranked higher in the school lottery than students who named the school as their second choice. That might seem compelling, reasoning that if the students say the school is their first choice, they are likely to 'want' the school more which might make them better students. It should also mean they are more likely to accept the schools offer than someone who makes the school its second choice.

This kind of argument ignores the fact that students who want to get into the school can increase the chances of being offered a seat by naming it as their first choice. This is true not just for the school the student really wants, but for all the others as well. If a school is very popular, many students will name it as their first choice. So any individual student has a small chance of being offered a seat. If the student really just wants into a program, it can sometimes do better by naming one of the other schools as its first choice.

At this point, we can run through a trivial example which will illustrate a lot of the problems.

There are around 3 students applying for each seat in the Vancouver school system. So lets imagine an example with 3 students trying to get into one of two schools, each of which has exactly 1 seat. Each student likes School 1 more than School 2, but they also like School 2.

It isn't reasonable to expect them all to name School 1 as their first choice. School 1 is popular, and they know they will be in a lottery for the seat in the school. They might win if they join the lottery. If they don't, they might get an offer from School 2 which won't be able to fill its seats with first choice students. However, it is also possible that they won't get any seat at all.

It is pretty easy to remedy that - instead of naming School 1 as their first choice, they just have to name School 2 as their top choice. They won't get School 1, but they will get a seat for sure at School 2, which they also like, because School 2 will give them priority for naming it as their top choice.

So there is no particular reason to believe that naming a school as the top choice means that it really is the student's top choice. This problem has another downside in that students have to waste their time trying to figure out how to insure a place in *some* school instead of thinking about which school suits them best. Apart from the sheer hassle of trying to figure this out, it is also stressful. It is easy to see that one school has a better teacher or principal, it is much harder to try to second guess what other students are going to list as their first choices.

This reflects one of the biggest advantage of the deferred acceptance method, since students don't have to think about what other students are doing at all - they can never improve on stating exactly which school is their most preferred school.

*How to bet on school choice.* To continue illustrating some of the problems, suppose as before there are two schools, School 1 and School 2, each with one seat. Again there are three students applying. Students $A$ and C like school 1, while student $B$ likes School 2 the most. Again, each of the students is willing to go to either of the two schools.

The students don't know any of this stuff, all they really know is that School 1 is more popular than School 2, and that to have any reasonable chance of getting into a school, they need to name it as their first choice. Suppose their payoff if they get into their most preferred school is 3, while the payoff to the less preferred school is 2. They might guess that the chance they get into school 1 is $\frac{1}{3}$ while the chance that they get into school 2 is $\frac{2}{3}$.

For student $B$, this makes life easy, because he *expects* a payoff of $\frac{2}{3}3 = 2$ by naming school 2 as his first choice. It is a bit harder for students A and C because they expect a payoff of $\frac{1}{3}3$ when they name school 1 as their first choice and $\frac{2}{3}2 = \frac{4}{3} > 1$ when they name school 2, so they will both name school 2 as their favorite choice - even though it isn't their favorite school, they have a better chance of getting in.

Now the second problem kicks in. Schools 1 and 2 run independent lotteries over the students who chose it first and the students who chose it second. It will very likely happen that student B ranks very low at School 2. Then either $A$ or $C$ will be given a seat in school 2 - apparently their first choice schools. Suppose $A$ gets the seat.

Now something quite odd happens. School 1 has no students who chose it as their most preferred school. So it should offer its seat to the top ranked student who made it their second choice. One possibility is that A is the student - a student who already has a seat in what he said was his first choice school. If the seat is offered to $A$,he will accept it - that is, reject his first choice in favor of his second choice. This weird event occurs because naming a school as your first choice increases your chances of getting in. This outcome is also very confusing for the person who is supposed to make the offer to student A.

Of course, the offer could also go to student B, so that both students are actually placed in their second choice schools. This outcome is not 'pareto optimal' because they could swap places and make themselves both better off without impacting $C$.