Algorithms for Foundation School Matching

Rob Irving and David Manlove

Computing Science Department, University of Glasgow

20 May 2009

This report summarises the procedure currently in use for matching applicants to Foundation Schools in the U.K., highlights the advantages and disadvantages of this approach, and considers some alternatives. In order to do this, some general principles that are relevant for centralised matching are introduced and discussed. Some results of an initial small-scale empirical study are presented. The report draws on the experience of the authors, over a number of years, in the design and implementation of the matching scheme used within SFAS, the Scottish Foundation Allocation Scheme [1], and on their considerable research expertise in algorithms for matching problems [2].

1. General concepts

Centralised schemes for matching applicants to employers arise in many real-world settings. In this section we will use the terms *applicant* and *employer* for consistency, although, in practice, "applicants" may be doctors, students, school pupils, etc., and "employers" may be hospitals, colleges, schools, etc. Also, for brevity, we refer throughout to an applicant as 'she' (to reflect the preponderance of female applicants in Foundation School matching).

The details of such schemes can vary considerably depending on the policies and objectives that apply in any particular context. In some settings, only applicants have preferences, in others, both applicants and individual schools have preferences, and in still others the preferences of employers are in accordance with a *master* preference list constructed on the basis of some scoring system applied to the applicants. A further distinction is between cases where all preference lists are strictly ordered and those where *tied* preferences are allowed, on one side or on both sides. For example, such ties arise naturally from scoring schemes where applicants can have identical scores. A further issue is whether an applicant is required to include all employers in her preference list (the list is said to be *complete*), or more commonly, only a chosen set of employers, perhaps of fixed size.

The term *matching* is used for an allocation of applicants to employers in which each applicant is allocated to at most one employer, and each employer is allocated a number of applicants that does not exceed the number of posts available, the *capacity* of the employer.

The question arises as to what properties make one matching better than another. What objectives are appropriate in any particular context? There is no unique answer to these questions, since desirable attributes may simply be unattainable, or may be mutually conflicting.

Among the properties that may be relevant are the following:

- The *size* of the matching
- The *profile* of the matching
- Whether the matching is *stable*
- Whether the matching is *exchange-free*
- Whether the matching mechanism is *strategy-proof*.

We consider each of these issues in turn.

Size

The *size* of a matching simply refers to the number of applicants actually matched. Not all applicants need be matched; this will be inevitable if the number of applicants exceeds the total number of available posts, but may also happen if preference lists are incomplete (since, in general, it is assumed that a matching scheme will not allocate an applicant to an employer that does not appear on her preference list).

Profile

The (applicant) *profile* of a matching specifies the numbers of applicants matched to their first, second, third . . . preference. So, for example, a profile of (35, 10, 4, 0, 2) would describe a matching in which 35 applicants achieve their first choice, 10 their second choice, 4 their third, no-one their fourth, and 2 their fifth, with no-one receiving worse than their fifth choice. It is not necessarily straightforward to decide whether one profile is better than another; different perspectives are possible – for example, on whether the priority is to maximise the number of first choices, or minimise the number of last choices, and so on. Further discussion of this issue appears in Section 3 of this report.

Stability

A matching is *stable* if there is no applicant A and employer E so that

- A is either unmatched or would prefer to be allocated to E, and
- E either has an unfilled place or A is a preferred applicant as compared to at least one of E's assignees.

It is apparent that the stability property is relevant only in contexts where applicants have preferences and employers have preferences, either explicitly and individually, or implicitly on the basis, say, of a master list of scores. The essence of stability is that it would be impossible for an applicant and employer to come to a private arrangement, outside the scheme, which would be to the advantage of both of them. Another perspective is the following: suppose that an applicant enquires why she was not allocated to a more preferred employer; if the matching is stable then there is a standard answer – "because all of your preferred employers filled all of their posts with applicants having scores at least as high as you". If the matching is not stable, then this answer cannot be given in at least some cases.

Studies of the level of success enjoyed by various matching schemes over many years have argued convincingly that stability is a key property [3, 4]. Schemes that produce stable matchings – prime examples include the National Resident Matching Program in the United States [5] – tend to be well regarded and to stand the test of time. On the other hand, those that produce unstable matchings – early matching schemes for medical students in Birmingham and Newcastle are well-known examples – sooner or later lose the trust of the community that they serve and become discredited.

The exchange-free property

A matching is *exchange-free* (sometimes called Pareto efficient, or Pareto optimal) if there is no set of two or more applicants who could arrange to exchange their assignments in some way such that all of them become better off. It turns out that if employers have individual preference lists, it might be impossible to guarantee simultaneously the stability and exchange-free properties. However, both properties can be achieved simultaneously when there is a master preference list of applicants.

The strategy-proof property

A matching algorithm is *strategy-proof* if there are no circumstances in which an applicant can improve her chances of a favourable outcome, or reduce the chances of an unfavourable outcome, by submitting a preference list that misrepresents her true preferences. This is an important property; if applicants believe or suspect that strategic behaviour may be profitable then respect for the matching process is likely to diminish.

Discrimination

One further desirable property of a matching, in most circumstances, is that of *discrimination*. There is no precise definition of this concept – it is an indication of whether the scheme discriminates, in at least some way, between applicants on the basis of their scores or rank ordering. For example, it would be possible to describe an algorithm that produces a matching with a profile that is guaranteed to be best possible in some precise sense, but this might not be acceptable if it involved treating all applicants on an equal basis, so that, say, the weakest applicant was as likely to achieve her first choice as the strongest one. In fact, it is not hard to see that requiring stability is one way of ensuring some level of discrimination.

2. The current Foundation School matching algorithm

The description given here of the algorithm currently employed to allocate applicants to Foundation Schools on the basis of their preferences and scores is adapted from [6] and [7]. This algorithm is henceforth referred to as Algorithm C (for Current). We now focus exclusively on the Foundation School context, so henceforth we use the term “school” rather than “employer”.

Algorithm C is essentially a hybrid of two well known algorithms, the so-called “First Preference First” and “Serial Dictatorship” methods.

Rather than a precisely defined algorithm, “First Preference First” is more properly an umbrella term for any algorithm that gives priority, in allocating places in a school, to applicants for whom that school is first preference. It is interesting to note that such algorithms are inherently non strategy-proof, and because of the strategic behaviour that such algorithms inevitably encourage among applicants (see below), the Department for Children, Schools and Families has recently outlawed the use of any such procedure in allocating children to school places in England. In fact, schools are obliged to offer places to pupils without knowing the pupils’ preference lists [8].

Algorithm C Phase 1

Phase 1 of Algorithm C proceeds essentially as follows. The list of applicants is ordered by score, with some random mechanism used to order applicants having equal scores. We refer to this mechanism as *tie-breaking*, since applicants with equal scores can be viewed as forming a tie in the original master list, and we call the resulting strictly order list the *rank ordering* of applicants. It is worth noting that, because of the rather coarse grained scoring system, many of the ties that have to be broken are very large, containing several hundred applicants. The rank ordered list is then processed by allocating each applicant in turn to her first choice school if a place at that school remains, and if not, deferring further consideration of the applicant to Phase 2. This is the “First preference first” aspect of Algorithm C, and it implies that if applicant A’s first choice school has *n* places, then she will not be assigned a place there if that school was first choice for *n* or more applicants higher in the rank ordering. Note that these *n* applicants might include a significant number with the same score as applicant A, but who were more fortunate than A at the tie-breaking stage.

It is worth noting that the outcome of Phase 1 – in terms of which applicants are assigned – depends not only on the applicant scores and preferences, and the number of places at each school, but also on the way in which the ties are broken. However, the number of places filled in a particular school during Phase 1 is not affected by tie-breaking; it is equal to the number of applicants for whom the school was first choice, or the capacity of the school, whichever is the smaller.

Algorithm C Phase 2

Phase 2 then deals with the deferred applicants, again in the order in which they appear in the rank ordering. But the mechanism in Phase 2 is quite different – it is the so-called "Serial dictatorship" approach. Each applicant in turn is assigned to a school, namely the one appearing highest on her preference list that still has at least one place remaining. By this means, every applicant will be assigned to some school, provided that the total number of places is at least equal to the number of applicants. This is a consequence of the fact that applicants' preference lists are complete, so that an applicant could end up being unassigned only if every school were already full. Again, the nature of allocations made during Phase 2 is dependent on the ways in which the ties are broken. This affects not only the outcome for individual applicants, but also the profile of the allocation. (The number of first choices will not be affected, but in general the rest of the profile will.)

Positive aspects of Algorithm C

Algorithm C has one notable positive property, namely that the maximum possible number of applicants are allocated to their first choice school. This is true regardless of the outcome of the tie-breaking stage. It also leads to a matching that is exchange-free (though, in reality, this property would be true of almost any reasonable scheme when a master preference list of applicants is involved.) However, the advantage of maximising first choices is achieved at a substantial cost. There are three crucial negative aspects of this algorithm.

Negative aspects of Algorithm C

First of all, and crucially, the outcome is not guaranteed to be stable; indeed there is likely to be a significant number of applicants who would prefer to be matched to some school or schools that have been assigned applicants with lower scores. As one example of how this can happen, consider an applicant A with a moderate score who puts a popular hospital X first on her list and a slightly less popular hospital Y in second place. It may well happen that when A is reached in Phase 1, all places at X have been allocated, so that A is deferred to Phase 2. But by the time A is reached in Phase 2, all places at Y may be filled, some of them by applicants, weaker than A, who happened to rank school Y in first place. As a result, during Phase 2, A cannot be allocated to her second choice and ends up with a worse assignment.

Furthermore, those applicants not realising their first choice are likely to fare considerably worse than under alternative schemes, and some applicants are likely to be matched to schools that are very low on their preference list. In other words, the profile of the matching will have decidedly undesirable features to offset the (admittedly desirable) number of first choices.

Finally, it is quite easy to appreciate that the algorithm is by no means strategy-proof. For example, it is a considerable risk for an applicant A with a modest score to put a popular school in first place; if that school is filled by applicants better than (or at least as good as) A, it is possible that the next several schools on A's preference list were also filled during Phase 1 of the algorithm, and some of the applicants filling these schools may be much weaker than A. It follows that, in deciding which school to place first on her preference list, there is a temptation for an applicant to attempt to "second guess" the behaviour of other applicants, and somehow to try to assess the relative popularity of schools. An applicant may decide to place first a desirable school that she believes to be attainable with some certainty, rather than the school that is her genuine first choice.

Not all of these disadvantages are immediately apparent by examining the statistical results of the 2008-09 run of the algorithm [9]. Lack of stability and the absence of the strategy-proof property cannot be discerned from the numbers alone. However, the undesirable features of the profile are immediately apparent. In fact, the profiles for the matchings obtained by Algorithm C on the 2008 and 2009 data were as follows [10]:

2008: (6498, 46, 56, 54, 102, 113, 68, 46, 36, 30, 15, 7, 1, 2, 0, 0, 1, 0, 0, 0)

2009: (6291, 184, 91, 53, 56, 73, 47, 37, 33, 19, 19, 23, 15, 14, 16, 14, 7, 4, 4, 4)

The patterns appear slightly different (for reasons that are not immediately apparent), but it is clear that in both cases a significant number of applicants were allocated to choices well down on their list – for instance, in 2009, 120 applicants failed to be matched to one of their top 10 choices.

(Of course, these figures may have to be taken with a pinch of salt. If at least some applicants are behaving strategically and misrepresenting their true first choice, then perhaps the number of first choice assignments here is really an over-estimate.)

It is worth emphasising that the lack of stability, in particular, has the potential to be a seriously damaging shortcoming. There must be some possibility that a successful legal challenge could be mounted by an applicant who was able to show that weaker applicants had been assigned to a school that she would have preferred. It seems likely that the prospect of legal challenges of this kind was a key reason that, as mentioned earlier, first preference first algorithms in schools matching were recently banned by the Department of Children, Schools and Families.

3. Possible alternative algorithms

Algorithm SD – pure serial dictatorship

The most obvious alternative to Algorithm C would involve abandoning Phase 1 and basing the entire approach on serial dictatorship. Let us consider Algorithm SD that proceeds in essentially the same way as Phase 2 of Algorithm C:

- First break all ties randomly to produce a strict rank ordering of the applicants;
- Process this rank ordered list one applicant at a time;
- Assign each applicant to the first school on her preference list that still has an available place.

Algorithm SD has two crucial advantages over Algorithm C, namely that it produces a stable matching, and it is strategy-proof. Also, like Algorithm C the matching is guaranteed to be exchange-free. It is not likely to allocate as many applicants to their first choice schools, but that may well be a price worth paying to achieve these other positive features.

The stability and strategy-proof property of Algorithm SD hold regardless of the outcome of the tie-breaking step. However, it is clear that the precise outcome, and in particular the allocation for many individuals, will depend on the actual ways in which ties are broken. Indeed, the profile of the resulting matching can also vary quite significantly, and it is difficult to predict what the profile might look like, and how it would compare with the profile resulting from application of Algorithm C, in any particular instance.

Comparing profiles

The question arises as to how we should compare matchings based on profile, and therefore, other things being equal, how we might choose, on the basis of profile, between various alternative matchings (that all might be known to be, say, stable and exchange-free).

Consider the following hypothetical profiles (based on a matching of size 7000):

> **Matching W** has profile (6000, 500, 200, 100, 50, 50, 50, 20, 20, 10).
>
> **Matching X** has profile (5500, 800, 300, 200, 100, 50, 50).
>
> **Matching Y** has profile (5800, 600, 200, 100, 100, 100, 100).
>
> **Matching Z** has profile (6000, 450, 150, 50, 100, 100, 50, 50).

How can we compare these profiles? Can any one of them be designated unequivocally as the best of the four?

In W and Z, more applicants receive their first choice, but in X and Y no applicant receives worse than her 7^{th} choice, whereas some applicants received their 8^{th} choice in Z and their 10^{th} choice in W.

There are various rules, or *optimality criteria*, that can be used to compare profiles, and it is up to those in charge of a matching scheme to decide which of these rules may be appropriate in any particular circumstances. We describe four such rules.

Criterion 1: the *Greedy* rule. The objective is to maximise the number of first choices, then subject to that, to maximise the number of second choices, then maximise the number of third choices, and so on. So, according to this criterion, matching W is the best of the four matchings defined above.

Criterion 2: the *Generous* rule. Suppose that there are m schools. The objective here is to minimise the number of m^{th} choices, then subject to that, to minimise the number of $(m\text{-}1)^{th}$ choices, and so on. So according to this criterion, matching X is the best of the four matchings defined above.

Criterion 3: The *Amended-generous* rule. The objective is to maximise the number of first choices, then subject to that, to minimise the number of m^{th} choices, then minimise the number of $(m\text{-}1)^{th}$ choices, and so on. So, according to this criterion, matching Z is the best of the four matchings defined above.

Criterion 4: The *Amended-greedy* rule. The objective is to have a profile that is the same length as the best possible generous profile, but subject to that, to maximise the number of 1^{st} choices, then maximise the number of 2^{nd} choices, and so on. So according to this criterion, matching Y is the best of the four matchings defined above.

Algorithm RSD – repeated serial dictatorship

Algorithm SD, in the form described, gives us no control whatever over the profile of the matching. When the algorithm is run, a matching is produced that depends on the ways in which the ties are broken in the rank ordered applicant list. This may give a better or worse profile, in terms of any of our optimality criteria, depending on the luck of the draw.

However, this does suggest a slightly more elaborate algorithm in which serial dictatorship is executed many times using different random tie-breaking each time. Among all the various matchings that are so generated, we keep the one that is best with respect to whichever optimality criterion we wish to use – any one of the four listed above, or indeed any other that we may wish to define.

The algorithm can be run as many times as we please, depending on the actual time that this takes. The more runs that are used the more chance there is of an improved profile.

As with the matching arising from a single run of Algorithm SD, the matching produced by Algorithm RSD is guaranteed to be exchange-free and stable. So the question arises whether there is any penalty to be paid for a likely improved profile. The answer is that Algorithm RSD is no longer truly strategy-proof. It turns out that, in some sense that is difficult to quantify, an applicant might, for example, be able to increase the chance of being allocated to her first choice by ranking popular schools highly, regardless of her true feelings for these schools. However, this is likely to be at worst a very marginal effect, not nearly as significant in reality as the strategic issues that arise with respect to Algorithm C.

"Clever tie-breaking" algorithms

More sophisticated algorithms are possible in this situation. For example, suppose that the requirement were to produce a stable, exchange-free matching which had the absolutely best possible profile with respect to a particular chosen optimality criterion. It turns out that this kind of computational problem is very challenging; for technical reasons, it is unlikely that an algorithm that is efficient, i.e., one that could be implemented to run in a reasonable time interval on moderate sized data sets, and that would guarantee to find the optimal solution, is possible.

However, what is possible is to design an efficient "heuristic" approach that should produce solutions close to optimal – most likely with a better profile than would be achieved by Algorithms SD or RSD. Currently, such a heuristic algorithm is used in SFAS to allocate applicants to Foundation Programmes in Scotland. In the Scottish context, the situation is slightly different. Programme directors construct individual preference lists, based on but not completely constrained by, the applicants' scores. In addition, preference lists are not complete – currently applicants submit a preference list of length 6 – so the main issue is to find a stable matching that allocates as many applicants as possible to one of their 6 choices, with profile being a secondary consideration. Experience over the past four years suggests that "clever tie-breaking" algorithms do succeed in matching significantly more applicants than random tie-breaking algorithms [11]. However, it is not clear whether such algorithms can be adapted to focus on finding stable matchings with profiles that are significantly better than those found, say, by Algorithms SD or RSD.

Finally, it has to be admitted that, like Algorithm RSD, these heuristics are not truly strategy-proof, though it is again difficult to analyse them from this standpoint. But there are certainly some circumstances in which an applicant, whose most preferred schools are popular, might improve her chances of being allocated to one of these by avoiding placing unpopular schools high on her preference list.

Abandoning stability / discrimination

In most circumstances, it seems natural that a matching scheme should involve some kind of discrimination in favour of stronger applicants, whether this manifests itself as a requirement for stability, or in some other way. However matching schemes do exist that are based solely on the applicants' preferences, with no employer preferences or scores of any kind. One that is known to the authors is the scheme that matches probationary teachers in Scotland to employing authorities, the so-called Scottish Government Teacher Induction Scheme [12].

When matching is to be done on the basis of applicant preferences alone, there are efficient, sophisticated algorithms (developed by the Algorithms group at the University of Glasgow) that can guarantee to match the maximum number of applicants and to produce the optimal profile according to any one of the optimality criteria listed earlier. These algorithms have been used successfully for several years in the Medical Faculty at the University of Glasgow to allocate students to options and placements [13].

Dealing with linked applicants

Nothing has been mentioned, hitherto, regarding the issue of linked applicants. It is assumed that linked applicants wish to be assigned to the same school, and that they submit identical preference lists. So a pair of linked applicants can be viewed as a special kind of applicant, a *couple*, that occupies two places in the allocated school. Just what score should be associated with a couple is a policy decision; currently it is effectively the lower of the scores of the two individuals involved. There are obvious alternatives, such as the average of these two scores.

Any matching algorithm that is used in this context must obviously be adapted to take account of the need to allocate two places to a couple. Indeed, the definitions of the stability and exchange-free properties are not so straightforward in this setting. The changes required to Algorithms SD and RSD are minimal. However, as seems to be recognised by the current scheme, there are some extreme circumstances in which it may not be possible for any particular algorithm to allocate both members of a couple to the same school. For example, a couple might be last in the rank ordering, and when Algorithm SD reaches that couple there might be no school that has more than one place available. So it seems inevitable that the scheme must have the right to break the link in such cases.

4. Some preliminary empirical evidence

The short time-scale for the preparation of this report allowed for only a few preliminary and small-scale computational experiments. Hence these should be taken only as an indication of the characteristics of the algorithms involved.

We were given access to the raw data from 2008 and 2009. This data contained withdrawn, ineligible and special circumstances applicants. However, rather than try to edit this data into a form that would have corresponded exactly to the data used by the matching program in these years, we decided just to proceed with this raw data. We also decided, in the interests of speed and simplicity, to ignore linked applicants. All of this means that the comparisons between the different outcomes on these data are perfectly valid, but these outcomes should not be compared directly with the actual results of the matching scheme in 2008 and 2009.

We implemented Algorithm RSD and an analogous version of Algorithm C – let us refer to it as Algorithm RC – that allowed for repeated executions, saving the "best" matching. We ran each algorithm for 100,000 repetitions, which took just a few seconds on a desktop PC. For each algorithm, we saved the matching with the best profiles according to the "greedy" and "generous" rules described in Section 3. In addition to the profile, we also computed two additional aspects of each matching:

- The number of *blocking pairs*; this is the number of applicant-school pairs (A, S) such that A prefers S to her assigned school and S has at least one assigned applicant with a smaller score than A. Hence this is a measure of instability – a stable matching is guaranteed to have no blocking pairs.
- The number of *blocking applicants*; this is the number of different applicants involved in a blocking pair, and is another important measure of instability.
- An indication of how "unlucky" applicants might be. For each rank *k* we recorded the highest score of an applicant assigned to her k^{th} preference. This can be seen as another indicator of discrimination, quite different in nature from stability.

Tables 1 and 2 show the profile and stability characteristics of matchings found by the algorithms on the raw 2009 and 2008 data respectively. The first row of each table refers to a single execution of Algorithm C, and subsequent rows to the best solutions obtained, in terms of profile, over 100,000 iterations of Algorithms RC and RSD. Note that the profiles for the 2009 data have 19 entries in all cases, and those for the 20o8 data all have 8 entries.

Tables 3 and 4 show, for a random solution produced by each of the two algorithms, the maximum score of an applicant assigned to her k^{th} preference school, for each value of *k* in the relevant range.

Time did not permit an implementation of a suitably tailored "Clever tie-breaking" algorithm aimed at finding stable matchings with favourable profiles. It is not clear whether such algorithms would lead to profiles substantially better than those found by Algorithm RSD.

Alg	Solution	Profile	No. blocking pairs *and applicants*		
			Min	Max	Ave
C	Random	(6267,177,84,47,61,73,46,37,29,12,24,21,13,16,9,8,2,1,2)	1322 *437*		
RC	Greedy	(6267,188,81,44,63,63,43,36,32,13,22,21,13,16,11,9,4,1,2)	1285	1417	1346
	Generous	(6267,177,84,44,68,64,44,37,33,18,21,25,14,13,10,8,1,0,1)	*425*	*450*	*437*
RSD	Greedy	(5998,414,148,65,70,71,47,27,24,18,14,8,3,6,8,6,0,0,1)	0	0	0
	Generous	(5982,430,145,65,67,77,49,23,27,20,16,8,2,4,10,2,0,0,1)	*0*	*0*	*0*

Table 1. Raw 2009 data. Profile and stability of various solutions

Algorithm	Solution	Profile	No. blocking pairs *and applicants*		
			Min	Max	Ave
C	Random	(6234,20,41,52,98,83,23,8)	848 *301*		
RC	Greedy	(6234,22,44,48,97,83,23,8)	835	866	850
	Generous	(6234,20,41,53,101,82,21,7)	*299*	*303*	*301*
RSD	Greedy	(5927,333,101,61,82,39,13,3)	0	0	0
	Generous	(5918,333,107,68,81,43,6,3)	*0*	*0*	*0*

Table 2. Raw 2008 data. Profile and stability of various solutions

Alg	Highest score of applicants assigned to each preference																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
RC	100	73	72	72	72	72	72	67	67	68	66	66	64	65	62	61	60	57	56
RSD	100	73	72	70	70	67	66	64	62	61	60	61	60	58	57	54	-	-	46

Table 3. Raw 2009 data. Indicator of the "unluckiest" applicants for various solutions

Alg	Highest score of applicants assigned to each preference							
	1	2	3	4	5	6	7	8
RC	94	71	71	71	71	69	68	59
RSD	94	71	64	60	57	55	55	48

Table 4. Raw 2008 data. Indicator of the "unluckiest" applicants for various solutions

Comments on the empirical results

Tables 1 and 2 confirm some of the observations made earlier in this report. In particular, Algorithm RC (like Algorithm C) always assigns the maximum possible number of applicants to their first choice post, significantly more than Algorithm CSD (or Algorithm SD).

Because of this feature, the profiles typically obtained by the two algorithms are not strictly comparable. Algorithm SD assigns fewer applicants to schools that are well down their list, but the "tail" of the profile, though smaller, is still present.

The number of blocking pairs, and more crucially, blocking applicants, is quite striking – over 400 blocking applicants in the 2009 case and around 300 for the 2008 data. So, typically, one might expect several hundred applicants to have a potentially valid grievance over a matching generated by Algorithm C.

Tables 3 and 4 illustrate another respect in which Algorithm C can treat at least some good applicants relatively badly. For example, on the 2009 data, some candidates with a score of 72 were assigned to their 7th choice, and some with a score of 65 to their 14th choice. With Algorithm SD no such applicant was assigned to worse than their 3rd and 7th choices, respectively.

Preferences and scoring schemes

The current scoring scheme appears to give scores in the approximate range 40 to 100. The question arises as to what effect a scoring mechanism with a finer, or a coarser, granularity might have on any matching scheme. This is difficult to predict with any precision, but a general principle is that imposing more constraints on the matching process is likely to have a deleterious effect on the matching profile. This implies that it is best to avoid a scoring scheme that is so fine that essentially artificial or random distinctions are made between applicants. This principle applies also to applicant preferences: is it really feasible for an applicant to provide a genuine strict ranking of over 20 schools? Allowing an applicant to have "ties" in her preference list would reduce the number of constraints in the system, and in principle might allow "better" matchings to be found, at least in cases where stability is a required property.

The next subsection gives an extreme illustration of how reducing (or in this case removing) constraints arising from discrimination can result in a dramatically improved profile.

What if scores are ignored altogether?

The two sets of raw data were also input to algorithms designed to consider only applicants' preferences, ignoring the scores completely (see earlier subsection "Abandoning stability/discrimination"). The results turned out to be quite striking. For both sets of data, it would have been possible to assign every applicant to either her first or second choice. Table 5 summarises the optimal profiles obtained, from the greedy and generous viewpoints, in the two cases. So, for example, on the 2009 data, the algorithm that found the generous optimal solution assigned 6127 applicants to their first choice and 804 to their second choice – but of course there is no telling just which applicants are in the two sets; weak applicants are as likely as strong applicants to get their first choice here.

It is not our intention to advocate this approach, but these results give a vivid indication of how different desirable objectives are, in fact, conflicting. There is no "silver bullet" that can lead to a solution that is ideal in all respects.

	2009	2008
Optimal greedy profile	(6267,541,115,8)	(6234,126,58,46,95)
Optimal generous profile	(6127,804)	(6004,555)

Table 5. Optimal profiles when scores are ignored.

5. Conclusion

In this report we have described some desirable properties of a matching that we may wish to achieve through use of an appropriate matching algorithm. However, we have illustrated that, in general, some of these properties may be mutually conflicting and it is likely to be impossible to satisfy all of them simultaneously.

We have discussed Algorithm C, the algorithm currently used by UKFPO. Its key advantage is that it maximises the number of first preference assignments, but this is achieved at considerable cost – lack of stability, problems with the overall matching profile, and the fact that its use is likely to lead to strategic behaviour among the applicants.

We have described a possible alternative to Algorithm C, namely Algorithm RSD, which essentially runs Phase 2 of Algorithm C for some fixed (large) number of iterations, keeping track of the matching which has the "best" profile according to some optimisation criterion. Although the time available for empirical investigation has been limited, preliminary computational evidence suggests that, whilst Algorithm RSD cannot be expected to produce as many first choices as Algorithm C, and may not completely eliminate the "long tail" that may be present in the matching profile generated by Algorithm C, its undoubted advantage is that the matching constructed will be stable. Moreover, as illustrated in the empirical results, Algorithm C is likely to treat some good applicants quite badly, a problem that arises to a much lesser extent with Algorithm RSD. This alternative algorithm also reduces the incentive for applicants to behave strategically when constructing their preference lists.

There remains the possibility of a more sophisticated heuristic (probably based on "clever tie breaking") that will produce a stable matching with a more favourable profile than those given by Algorithms C and RSD. However, the development of such a heuristic represents a complex technical challenge, and, in reality, it is unclear how much better we can expect the profile to be, with typical data sets, if stability is prioritised.

Ultimately, the prioritisation of the various desirable characteristics is a decision for policy makers, given that we can never expect a single matching to be simultaneously optimal in all respects.

References

1. SFAS website, http://www.nes.scot.nhs.uk/sfas.
2. http://www.optimalmatching.com/
3. A. E. Roth, The evolution of the labor market for medical interns and residents: a case study in game theory, *Journal of Political Economy*, 92, 1984, 991-1016.
4. A. E. Roth, A natural experiment in the organization of entry level labor markets: regional markets for new physicians and surgeons in the U.K., *American Economic Review*, vol. 81, June 1991, 415-440.
5. NRMP web site, http://www.nrmp.org.
6. Web document http://www.foundationprogramme.nhs.uk/pages/medical-students/faqs
7. P. Brown, MMC Application Service: F1 Allocation Rules 2009, August 2008.
8. Department for Children, Schools and Families, School Admissions Code, 2009, http://www.dcsf.gov.uk/sacode/.
9. R. Palmer, Foundation programme 2009 recruitment, PowerPoint presentation.
10. C. McManus, Report on data received from UKFPO for 2008 and 2009 matching scheme, manuscript, May 2009.
11. R.W. Irving and D.F. Manlove, Finding large stable matchings, to appear in ACM *Journal of Experimental Algorithms*, 2009.
12. Scottish Executive Teacher Induction Scheme 2008-09, http://www.scotland.gov.uk/Resource/Doc/200891/0053701.pdf.
13 VALE, the Glasgow Medical School Virtual Administration and Learning Environment Portal, http://www.medicine.gla.ac.uk/.